UNIFIEDSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models

Tianbao Xie∗ 1, Chen Henry Wu+ 2, Peng Shi3, Ruqi Zhong4, Torsten Scholak5, Michihiro Yasunaga6, Chien-Sheng Wu7, Ming Zhong8, Pengcheng Yin9, Sida I. Wang10, Victor Zhong17, Bailin Wang11, Chengzu Li12, Connor Boyle17, Ansong Ni13, Ziyu Yao14, Dragomir Radev13, Caiming Xiong7, Lingpeng Kong11,12, Rui Zhang15, Noah A. Smith16,17, Luke Zettlemoyer10,17, Tao Yu1,17

1 The University of Hong Kong  2 Carnegie Mellon University  3 University of Waterloo  4 University of California, Berkeley  5 ServiceNow Research  6 Stanford University  7 Salesforce Research  8 UIUC  9 Google Research  10 Facebook AI Research  11 University of Edinburgh  12 Shanghai AI Lab  13 Yale University  14 George Mason University  15 Penn State University  16 Allen Institute for Artificial Intelligence  17 University of Washington

Abstract

Structured knowledge grounding (SKG) leverages structured knowledge to complete user requests, such as semantic parsing over databases and question answering over knowledge bases. Since the inputs and outputs of SKG tasks are heterogeneous, they have been studied separately by different communities, which limits systematic and compatible research on SKG. In this paper, we overcome this limitation by proposing the UNIFIEDSKG framework, which unifies 21 SKG tasks into a text-to-text format, aiming to promote systematic SKG research, instead of being exclusive to a single task, domain, or dataset. We use UNIFIEDSKG to benchmark T5 with different sizes and show that T5, with simple modifications when necessary, achieves state-of-the-art performance on almost all of the 21 tasks. We further demonstrate that multi-task prefix-tuning improves the performance on most tasks, largely improving the overall performance. UNIFIEDSKG also facilitates the investigation of zero-shot and few-shot learning, and we show that T0, GPT-3, and Codex struggle in zero-shot and few-shot learning for SKG. We also use UNIFIEDSKG to conduct a series of controlled experiments on structured knowledge encoding variants across SKG tasks. UNIFIEDSKG is easily extensible to more tasks, and it is open-sourced at https://github.com/hkunlp/unifiedskg.1

1 Introduction

Structured knowledge (e.g., web tables, knowledge graphs, and databases) stores large amounts of data in organized structures, forming a basis for a wide range of applications, e.g., medical diagnosis, personal assistants, and customer relations management. Accessing and searching data in structured knowledge typically requires mastering query languages through professional training. To promote the efficiency of data access, structured knowledge grounding (SKG) systems ground user requests in structured knowledge and produce various outputs, including computer programs (e.g., SQL and SPARQL), table cell values, and natural language responses (Figure 1). For example, semantic parsing (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005) converts natural language questions into formal programs; knowledge-base question answering (Berant et al., 2013) derives answers from tables or knowledge graphs.

SKG has attracted significant interest and has been studied through different tasks defined by different communities. Recent developments in tasks, models, and datasets for SKG have led to task-specific modeling advances, making each task’s progress seemingly unique and incompatible. A main reason is that SKG tasks are heterogeneous. Different types of structured knowledge, such as databases or knowledge graphs, lead to highly specialized encoders (Lin et al., 2019; Herzig et al., 2020; Wang et al., 2020; Yasunaga et al., 2021). Some SKG tasks, e.g., semantic parsing, use customized decoders to generate programs (Yin and Neubig, 2018; Ren et al., 2021). Therefore, instead of solving common challenges in SKG research, improvements in SKG have been prone to be exclusive to a single task, domain, or dataset.

In this paper, we propose the UNIFIEDSKG framework to advocate for a unifying view of 21 SKG tasks across six task families and multiple data domains (Table 1). UNIFIEDSKG standardizes datasets, models, code, experiments, and evaluation metrics into a single framework. By casting user requests, structured knowledge, and outputs
Figure 1: Structured knowledge grounding (SKG) leverages structured knowledge to complete user requests. By casting inputs and outputs into the text-to-text format, UNIFIEDSKG standardizes datasets, models, code, experiments, and metrics for 21 SKG tasks.

UNIFIEDSKG is easily extensible to more SKG tasks, and it is open-sourced to promote community-wide progress.

Using UNIFIEDSKG as a benchmark, we show that finetuning T5 (with constrained decoding or reranking when necessary) on individual tasks achieves state-of-the-art (sota) results on almost all of the 21 tasks, establishing a powerful and reproducible starting point for SKG research. T5 performance also increases with size on most tasks.

UNIFIEDSKG facilitates multi-task learning on SKG, enabling knowledge sharing and cross-task generalization. Although simple multi-task learning has mixed results, we show that multi-task learning with prefix-tuning (Li and Liang, 2021) benefits most tasks and largely improves the overall performance, on both T5-base and T5-large.

UNIFIEDSKG is a challenging testbed for few-shot (Brown et al., 2020; Ye et al., 2021a) and zero-shot learning (Zhong et al., 2021; Wei et al., 2021; Sanh et al., 2021) with PLMs. Our experiments show that models like T0 (Sanh et al., 2021) struggle in few-shot learning on SKG tasks, and GPT-3 (Brown et al., 2020) and Codex (Chen et al., 2021a) struggle in few-shot learning on SKG tasks.

UNIFIEDSKG enables a series of controlled experiments on structured knowledge encoding. We find that T5 is sensitive to encoding variations, and the sensitivity varies across tasks. UNIFIEDSKG aims to facilitate more general and robust structured knowledge encoding methods. Finally, we conduct a comprehensive error analysis across SKG tasks. Although the errors made by PLMs decrease with the model size, T5-3B may still generate invalid outputs.

In summary, we 1) unify and benchmark 21 SKG tasks under the UNIFIEDSKG framework to evaluate diverse grounding goals and structured knowledge sources, 2) demonstrate (near) sota performance of T5 on all the unified SKG tasks, using a single, general-purpose approach, 3) show the benefit of knowledge sharing across SKG tasks via multi-task prefix-tuning, and 4) analyze recent modeling contributions (zero-shot, few-shot, and structured knowledge encoding) on these tasks. We hope UNIFIEDSKG enables the design of new models and learning algorithms that generalize to diverse SKG tasks and to identify their challenges.

2 Related Work

SKG with PLMs PLMs have been applied to several SKG tasks. To encode structured knowledge, prior work linearized the structured knowledge and concatenated it with the text (Hwang et al., 2019; Liu et al., 2020; Hosseini-Asl et al., 2020; Liu et al., 2021), which has been augmented by positional encoding (e.g., row/column embedding) (Herzig et al., 2020; Yin et al., 2020a) and template-based linearization (Chen et al., 2020a,b; Oguz et al., 2021), and planning (Su et al., 2021). Recently, cell-column alignment is modeled by manipulating...
<table>
<thead>
<tr>
<th>Task Family</th>
<th>Task</th>
<th>Knowledge Input</th>
<th>User Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic Parsing</td>
<td>Spider (Yu et al., 2018)</td>
<td>Database</td>
<td>Question</td>
<td>SQL</td>
</tr>
<tr>
<td></td>
<td>GrillQA (Gu et al., 2021)</td>
<td>Knowledge Graph</td>
<td>Question</td>
<td>s-Expression</td>
</tr>
<tr>
<td></td>
<td>WebQSP (Yih et al., 2016)</td>
<td>Knowledge Graph</td>
<td>Question</td>
<td>s-Expression</td>
</tr>
<tr>
<td></td>
<td>MTOP (Li et al., 2021)</td>
<td>API Calls</td>
<td>Question</td>
<td>TOP Representation</td>
</tr>
<tr>
<td>Question Answering</td>
<td>WikiSQL (Zhong et al., 2017)</td>
<td>Table</td>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td></td>
<td>WikiTQ (Pasupat and Liang, 2015)</td>
<td>Table</td>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td></td>
<td>CompWebQ (Talmor and Berant, 2018)</td>
<td>Knowledge Graph</td>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td></td>
<td>HybridQA (Chen et al., 2020c)</td>
<td>Table + Text Passage</td>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td></td>
<td>MultiModaQA (Talmor et al., 2021)</td>
<td>Table + Text + Image</td>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td></td>
<td>FeTaQA (Nan et al., 2021a)</td>
<td>Table</td>
<td>Question</td>
<td>Free-Form Answer</td>
</tr>
<tr>
<td>Data-to-Text</td>
<td>DART (Nan et al., 2021b)</td>
<td>Triple</td>
<td>None</td>
<td>Text</td>
</tr>
<tr>
<td></td>
<td>ToTo (Parikh et al., 2020)</td>
<td>Highlighted Table</td>
<td>None</td>
<td>Text</td>
</tr>
<tr>
<td>Conversational</td>
<td>MultiWoZ (Budzianowski et al., 2018)</td>
<td>Ontology</td>
<td>Dialog</td>
<td>Dialog State</td>
</tr>
<tr>
<td></td>
<td>KVRET (Eric et al., 2017)</td>
<td>Table</td>
<td>Dialog</td>
<td>Response</td>
</tr>
<tr>
<td></td>
<td>SParC (Yu et al., 2019b)</td>
<td>Database</td>
<td>Multi turn</td>
<td>SQL</td>
</tr>
<tr>
<td></td>
<td>CosSQL (Yu et al., 2019a)</td>
<td>Table</td>
<td>Multi turn</td>
<td>Answer</td>
</tr>
<tr>
<td></td>
<td>SQA (Iyyer et al., 2017)</td>
<td>Table</td>
<td>Multi turn</td>
<td>Answer</td>
</tr>
<tr>
<td>Fact Verification</td>
<td>TabFact (Chen et al., 2020b)</td>
<td>Table</td>
<td>Statement</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td>FEVEROUS (Aly et al., 2021)</td>
<td>Table + Text</td>
<td>Statement</td>
<td>Boolean</td>
</tr>
<tr>
<td>Formal-Language-to-Text</td>
<td>SQL2Text (Shu et al., 2021)</td>
<td>Optional Database</td>
<td>SQL</td>
<td>Text</td>
</tr>
<tr>
<td></td>
<td>Logic2Text (Chen et al., 2020d)</td>
<td>Table Schema</td>
<td>Python-like program</td>
<td>Text</td>
</tr>
</tbody>
</table>

Table 1: We unify 21 SKG tasks with different knowledge input, user input, and output, covering six task families.

the attention matrix of transformers (Zhang et al., 2020; Eisenschlos et al., 2021). Hierarchical encoding is another way to represent the structure, e.g., Wang et al. (2021b) used tree-based transformers to represent the structure of the tables; Iida et al. (2021) used transformers to encode row and column representations; Chen et al. (2021b) used hierarchical transformers to encode KG triples. SKG’s outputs include, but are not limited to, structured meaning representations (e.g., logic forms, SQL), dialogue states, natural language, answer sets, and Boolean values. Among them, structured meaning representation is challenging for PLMs because they are originally trained on natural language. To bridge this gap, Shin et al. (2021) adopted the insights from Berant and Liang (2014) and Mar佐ev et al. (2020) and proposed to convert formal language into an English-like representation, decode with GPT-3, and map back to formal language automatically. We do not focus on these techniques in this work; instead, we unify all tasks and systematically compare them.

### Task format unification

Recent years witnessed the trend of unifying related but different tasks into a shared format. McCann et al. (2018) unified various tasks as question answering. Yin et al. (2020b) and Wang et al. (2021a) unified few-shot learning as textual entailment. PLUR (Chen et al., 2021c) unified program learning, understanding, and repair tasks into a graph-to-sequence format. In this paper, we focus on the text-to-text format (Raffel et al., 2020) due to its flexibility. Different from unifying tasks that only take text as input, a core challenge in unifying SKG tasks into the text-to-text format is to linearize structured knowledge. Notably, UniﬁedQA (Khashabi et al., 2020) uniﬁed QA tasks, while UNIFIEDSKG covers a broader scope of six task families for systematic exploration.

### Cross-task generalization with PLMs

Multi-task learning and transfer learning go beyond task boundaries, view different tasks as related, and have been shown to outperform single-task learning (Aghajanyan et al., 2021a; Vu et al., 2021). Large PLMs show potential for zero-shot and few-shot learning, e.g., GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), which can be improved by multi-task learning (Zhong et al., 2021), e.g., FLAN (Wei et al., 2021), T0 (Sanh et al., 2021), and CrossFit (Ye et al., 2021a). ExT5 (Aribandi et al., 2021) shows that scaling up multi-task learning helps improve pretraining efficiency and downstream performances. UNIFIEDSKG facilitates the investigation of multi-task, zero-shot, and few-shot learning on SKG tasks.

### 3 The UNIFIEDSKG Framework

#### 3.1 Task Unification

The guiding principle of UNIFIEDSKG’s task selection is diversity. We unify 21 SKG tasks across six task families and multiple domains (Table 1). Our task families include:

- **Semantic parsing** converts questions to logical forms (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005).
Figure 2: We unify SKG tasks with heterogeneous inputs and outputs into the text-to-text format.

- **Question answering** derives answers to natural language questions based on structured data (B erant et al., 2013).
- **Data-to-text generation** describes structured data in natural language (Novikova et al., 2017).
- **Fact verification** checks if a statement is true based on the structured data (Chen et al., 2020b).
- **Conversational tasks** require understanding of not only the user’s last request but also the full interaction history between users and machines (Budzianowski et al., 2018; Eric et al., 2019; Yu et al., 2019a).
- **Formal language to text translation** describes formal language in natural language (Chen et al., 2020d).

All these tasks take as input $x$ a user request, a structured knowledge input, and an optional (dialogue) context to predict an output $y$. Figure 2 illustrates how we convert the input $x$ to an input sequence $\tilde{x}$ and the output $y$ to an output sequence $\tilde{y}$ by means of “linearization” (Liu et al., 2021), enabling the unification of diverse forms of structured knowledge. We provide more details, examples, and input length analysis in the Appendices F and G. Our code implementation uses Hugging Face’s Transformers (Wolf et al., 2020) and Datasets (Lhoest et al., 2021) toolkits.

### 3.2 Modeling

The simplest usage of UNIFIEDSKG is to train on individual tasks. In this case, we minimize the negative log-likelihood loss averaged over tokens in each batch. For decoding, we use beam search by default. UNIFIEDSKG also facilitates exploration of multi-task learning, few-shot, and zero-shot learning with PLMs, and details are presented in the corresponding parts in Section 4.

### 4 Experiments and Analysis

#### 4.1 Results on Individual Tasks

We apply T5 models (Raffel et al., 2020) on each individual task in UNIFIEDSKG. For model training, we set the maximum number of epochs as 50–200, depending on the dataset size. We use early stopping and model selection on the development set. More details are shown in Appendix D.1. For each task, we report one commonly used metric in Table 2. See Appendix B for all metrics.

**Comparison with previous sota** Table 2 shows that vanilla T5-3B outperforms most previous sota models not trained on extra unsupervised in-domain data. Some semantic parsing sota models, denoted as + in Table 2, are also T5 with constrained decoding (Scholak et al., 2021) or reranking (Ye et al., 2021b). This shows that a generalist architecture like T5, when scaled up to a certain size, can be as good as task-specific architectures for SKG, suggesting the potential of larger PLMs.

**Model scalability** In general, T5 performance increases with the model size, but this trend varies across task families. Semantic parsing, QA, and fact verification tasks get large benefits from increased sizes, while text generation does not. See Section 4.5 for a human evaluation for text generation tasks. Also, the gap between T5-base (220M) and T5-large (770M) is larger than the gap between T5-large (770M) and T5-3B (3B).

**Effect of pretraining on structured knowledge** Some smaller models pretrained on structured knowledge (Liu et al., 2021) show competitive performance as T5-3B, suggesting that pretraining with structured data is beneficial for SKG. This result calls for structured knowledge pretraining that generalizes to different SKG tasks across domains, which can be systematically explored using UNIFIEDSKG.

### 3.2 Modeling

The simplest usage of UNIFIEDSKG is to train on individual tasks. In this case, we minimize the negative log-likelihood loss averaged over tokens in each batch. For decoding, we use beam search by default. UNIFIEDSKG also facilitates exploration of multi-task learning, few-shot, and zero-shot learning with PLMs, and details are presented in the corresponding parts in Section 4.
<table>
<thead>
<tr>
<th>Metric</th>
<th>T5-base</th>
<th>T5-large</th>
<th>T5-3B</th>
<th>Previous sota (w/o extra)</th>
<th>Previous sota (w/ extra)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spider (dev.)</td>
<td>Match</td>
<td>58.12</td>
<td>66.63</td>
<td>71.76</td>
<td><strong>75.5</strong> * (Scholak et al., 2021)</td>
</tr>
<tr>
<td>GraiQA</td>
<td>Match</td>
<td>62.39</td>
<td>67.30</td>
<td>70.11</td>
<td><strong>83.8</strong> * (Ye et al., 2021b)</td>
</tr>
<tr>
<td>WebQSP</td>
<td>F1</td>
<td>78.83</td>
<td>79.45</td>
<td>80.70</td>
<td><strong>83.6</strong> * (Ye et al., 2021b)</td>
</tr>
<tr>
<td>MTOP</td>
<td>Match</td>
<td>85.49</td>
<td>86.17</td>
<td>86.78</td>
<td>86.36 (Papapetrou et al., 2021)</td>
</tr>
</tbody>
</table>

| WikiTQ                 | Acc     | 35.76    | 43.22 | 49.29                     | 44.5 (Wang et al., 2019) | — |
| WikSQL                 | Acc     | 82.63    | 84.80 | 85.96                     | 85.8 (Liu et al., 2021) | 89.5 (Liu et al., 2021) |
| CompWebQ               | Acc     | 68.43    | 71.38 | 73.26                     | 70.4 * (Das et al., 2021) | — |
| HybridQA (dev.)        | Acc     | 54.07    | 56.95 | 59.41                     | 60.8 * (Eisenschlos et al., 2021) | 63.4 * (Eisenschlos et al., 2021) |
| MultiModalQA (dev.)    | F1      | 75.51    | 81.84 | 85.28                     | 82.7 (Yoran et al., 2021) | 83.8 (Yoran et al., 2021) |
| FeTaQA                 | BLEU    | 29.91    | 32.45 | 33.44                     | 30.54 (Nan et al., 2021a) | — |
| Dart                   | BLEU    | 46.22    | 46.89 | 46.66                     | 46.89 (Nan et al., 2021b) | — |
| ToTTo (dev.)           | BLEU    | 48.29    | 48.95 | 48.95                     | 48.95 (Kale and Rastogi, 2020) | 47.2 (Aghajanyan et al., 2021b) |
| MultiWoZ.2.1           | Joint Acc | 54.64 | 54.45 | 55.42                     | 60.61 * (Dai et al., 2021) | — |
| KVRET                  | Micro F1 | 66.45 | 65.85 | 67.88                     | 63.6 (Gou et al., 2021) | — |
| SPAR (dev.)            | Match   | 50.54    | 56.69 | 61.51                     | 54.1 (Hui et al., 2021) | 62.2 (Yu et al., 2021) |
| CoSQL (dev.)           | Match   | 42.30    | 48.26 | 54.08                     | 56.9 * (Scholak et al., 2021) | 52.1 (Yu et al., 2021) |
| SQA                    | Overall Acc | 52.91 | 61.28 | 62.37                     | 58.6 (Liu et al., 2021) | 74.5 (Liu et al., 2021) |
| TabFact                | Acc     | 76.13    | 80.85 | 83.68                     | 74.4 (Yang et al., 2020) | 84.2 (Liu et al., 2021) |
| FEVEROUS (dev.)        | Acc     | 75.05    | 79.81 | 82.40                     | 82.38 (Aly et al., 2021) | — |
| SQL2Text               | BLEC    | 93.52    | 93.68 | 94.78                     | 93.7 (Shu et al., 2021) | — |
| Logic2Text             | BLEC    | 90.66    | 90.57 | 91.39                     | 88.6 (Shu et al., 2021) | — |

Table 2: Test or development (dev.) set performance of models trained on individual tasks. Vanilla T5 or T5 with simple modifications (e.g., + constrained decoding or reranking) achieve sota on nearly all tasks. The best result without extra pretraining is shown in bold. More detailed results and result variances can be found in Tables 11 and 12 in Appendix. Human evaluation for generation tasks is in Section 4.5. w/ (w/o) extra means with (without) extra pretraining on unsupervised structured data (e.g., web tables).2

<table>
<thead>
<tr>
<th>Spider WikiTQ DART MWoZ TabFact SQL2Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5-3B 71.76 50.65 50.38 58.46 83.97 92.71</td>
</tr>
<tr>
<td>T0-3B 68.09 50.62 50.16 60.20 85.51 92.93</td>
</tr>
</tbody>
</table>

Table 3: Comparison between T5-3B and T0-3B. T0-3B is initialized from LM-adapted T5 and further pretrained on a large number of non-SKG tasks. We fine-tune both models on individual tasks. T0-3B under-performs T5-3B on semantic parsing (Spider) and outperforms T5-3B on dialogue state tracking (MWoZ) and fact verification (TabFact). We report results on the dev. set.

**Effect of pretraining on non-SKG tasks** T0-3B (Sanh et al., 2021) is initialized from T5-3B and pretrained on multiple tasks that (in most cases) do not use structured knowledge as input (non-SKG tasks). Exploring the performance of T0-3B on SKG tasks helps us understand the relationship between SKG tasks and non-SKG tasks. Table 3 shows that T0-3B under-performs T5-3B on semantic parsing and outperforms T5-3B on dialogue state tracking and fact verification. We note that T0-3B is pretrained on dialogue QA, dialogue summarization, and NLI tasks; therefore, pretraining on non-SKG tasks might not be useful for SKG unless we add similar SKG tasks to pretraining.

---

2 For GraiQA and WebQSP, we run T5 and rerun the previous sota model (Ye et al., 2021b) using the gold entities. For

4.2 Multi-Task Learning

**UNIFIEDSKG** facilitates the exploration of multi-task learning. In this part, we systematically study multi-task learning on all 21 unified tasks. We find that SKG benefits from multi-task prefix-tuning on both T5-base and T5-large, showing that the benefits from multi-task learning is scalable in terms of the model size. The baselines we use include:

- **Single-task finetuning (ST-F)**, which is finetuning on individual tasks, same as Section 4.1.
- **Single-task prefix-tuning (ST-P)**; Li and Liang, 2021, which learns lightweight task-specific pa...
parameters while keeping the PLM fixed. We set the prefix length as 10. Clive et al. (2021) also used prefix-tuning on T5 for data-to-text generation.

**Multi-task finetuning (MT-F)**, which combines the training data of all tasks with temperature mixing (Raffel et al., 2020); after hyperparameter tuning with a few steps, we set the temperature as 2). We select model weights based on the average metric on all tasks’ development set.

Table 4 shows that ST-P is comparable to ST-F on nearly all tasks. However, we find that it takes about 5–10 times as many training steps (See Appendix E), which is similarly observed for prompt-tuning (Lester et al., 2021). We also observe that MT-F leads to mixed results. For many tasks, MT-F is even worse than ST-F.

**Multi-task prefix-tuning (MT-P)** Our explanation for the mixed results of MT-F is that the inputs of SKG tasks contain different structured knowledge from diverse domains, making it difficult to learn shared parameters effectively. To address this challenge, we first pretrain a prefix on all tasks, freezing T5 and using the same temperature mixing as MT-F. In the second step, we initialize each task’s prefix with this pretrained prefix and optimize the prefix while freezing T5. This initialization step is similar to the prompt transfer explored in Vu et al. (2021). Following ST-P, we set the prefix length as 10.

Table 4 shows that multi-task prefix-tuning outperforms single-task finetuning and single-task prefix-tuning on most tasks, and it largely outperforms the naive multi-task learning baseline. It demonstrates that SKG tasks can be studied together to share data and knowledge.

**Exploring task knowledge transfer** UNIFIEDSKG facilitates studying knowledge transfer between SKG tasks. Given two tasks, task A and task B, we first train the model on task A and then continue training on task B. Table 5 shows that tasks benefit from other tasks with the same data source (e.g., tasks that all use Wikipedia tables as structured knowledge). We do not observe positive transfer between parallel tasks (e.g., semantic parsing tasks with different structured knowledge and different output) and subtask (e.g., question answering can be viewed as the execution semantic parses) when data sources are different. Compared to the positive results in Table 4, results in this part indicate that manually selecting source and target tasks may not be efficient for multi-task learning.

### 4.3 Zero-Shot and Few-Shot Learning

The text-to-text unification of UNIFIEDSKG enables us to investigate zero/few-shot learning on SKG with large PLMs.

#### Zero-shot learning setting

Zero-shot learning enables models to solve tasks with natural language descriptions without training samples. We follow T0 (Sanh et al., 2021) to create similar natural language instructions for the unseen tasks. Our instructions are provided in Appendix D.3.

#### Few-shot learning settings

Brown et al. (2020) showed that large PLMs could be few-shot learners.
by encoding a few training samples as “context” to learn without gradient updates. We use GPT-3 (Brown et al., 2020) and Codex (Chen et al., 2021a) to explore such few-shot learning for SKG. To stay within our budget, for GPT-3, we report the performance on 100 random dev. set samples. We explore two settings for few-shot learning.

In the first setting, we randomly sample few-shot examples from the training set; these examples are shared by all dev. set samples, denoted as random in Table 6. For sequences that are too long for Codex (4096) and GPT-3 (2048), we use as many examples as possible and make sure that there is at least one example (truncated if needed).

In the second setting, we follow Gao et al. (2021) to select few-shot examples from the training set. We call this setting few-shot with example selection, denoted as select in Table 6. We use the pretrained SBERT (Reimers and Gurevych, 2020) for sentence embeddings of the user request input (for tasks that only have structured input, we embed the linearized structured input) and sample five most similar examples measured by cosine similarity. Further details (e.g., prompts and task instructions) are provided in Appendix D.4.

SKG is challenging for zero/few-shot learning. Table 6 shows that zero-shot performance is very poor on most tasks (Spider and MultiWoZ are even 0). It also shows a large gap between few-shot learning and finetuning for Spider, WikiTQ, MWoZ, and TabFact, while the gap is smaller for generation tasks. For few-shot learning, example selection based on similarity outperforms random selection, but the gap is usually smaller than 10 points out of 100. It is also interesting to compare the results between synthesis tasks (Spider), which requires predicting programs, and induction tasks (WikiTQ and TabFact), where a model directly outputs answers (Devlin et al., 2017). We find that PLMs generally struggle more when adapting to induction tasks (e.g., close to random-guess on the binary classification task TabFact), reminiscent of recent attempts in program synthesis and induction using PLMs (Austin et al., 2021). For GPT-3 and Codex, better zero-shot performances can be expected by better prompt design.

### 4.4 Structured Knowledge Encoding

Structured knowledge encoding has been widely explored (Bogin et al., 2019; Lin et al., 2019; Agarwal et al., 2020; Saxena et al., 2020; Yasunaga and Liang, 2020; Yasunaga et al., 2022; and others detailed in Section 2). We hope that UNIFIEDSKG can promote systematic study of general structured knowledge encoding. To this end, this part focuses on the linearization of structured knowledge.

**Does the order of user input, structured knowledge, and context matter?** To explore the effect of the order of user input, structured knowledge, and context, we rerun the single-task experiments while switching the order of these components in both the training and development set. Table 7 shows that placing the text before structured knowledge (rs) is better than the opposite (sr), which is consistent across SKG tasks. Our explanation is that the position of the text is relatively fixed in rs,
helping the decoder to learn stable attention over the text. Also, placing the context in between the text and structured knowledge yields better results.

**Is T5 sensitive to structured knowledge ordering?** Order-insensitivity is common for most structured knowledge, e.g., permutation of columns in a table preserves the meaning. To study this insensitivity, we evaluate T5-large on a manipulated development set where the order of schema (for database), column (for table), or slots and values (for ontology) is reversed. Table 8 shows that tasks with cross-domain tables and databases are less order-sensitive, while models are very sensitive to the order of ontology. Other types of robustness (e.g., robustness to cell values irrelevant to the answer) remain an open question in UNIFIEDSKG.

**Is it beneficial to represent structured knowledge as natural language?** SKG data is not typically used to pretrain PLMs. Given ample training data, PLMs adapt well to SKG tasks, as shown in Table 2. However, under the low-resource setting, converting structured data to natural language might be helpful. For Spider, we use a shared template to convert structured data to natural language. For TabFact and WikiSQL, we randomly selected 236 tables shared by both datasets and manually labeled templates to convert each row into a sentence. Examples of the templates are shown in Appendix I. These templates produce about 1000 samples for each task, divided into training and test sets. We find that, in WikiSQL, the conversion to natural language stabilizes and accelerates the training process. Table 9 shows that conversion to natural language improves the performance on WikiSQL, has no significant influence on TabFact, and slightly degrades the performance on Spider.

### 4.5 Human Evaluation for Generation Tasks

For each generation task, we randomly sample 100 development set samples and ask human annotators to judge the correctness of each output, using a 0-1 score. Details are provided in Appendix D.5. Table 10 shows that automatic metrics do not always reflect human evaluation, calling for better automatic metrics to truly reflect the model’s ability on generation tasks. Larger models are not always better, and detailed error analysis is provided below.

#### 4.6 Error Analysis

**Error analysis based on output validity** Unconstrained decoding from PLMs may generate invalid outputs. For semantic parsing, we divide wrong outputs into invalid outputs (i.e., not executable when the output is SQL, and not parse-able when the output is s-expression or TOP-representation) and valid but wrong answers. Figure 3 shows that, for SQL semantic parsing, a large number of errors are caused by invalid outputs, and the number of invalid outputs gradually decreases with the increase of model size. This phenomenon is also observed by Scholak et al. (2021), who used constrained decoding to improve the validity, largely improving the parsing performance. For s-expression semantic parsing, invalid outputs take up 30–50% of all wrong outputs, and increasing the model size does not reduce invalidity significantly. For fact verification tasks, valid outputs are “entailed” and “refuted”. We observe that T5 always generates valid outputs. For question answering, we do not include the validity analysis since the validity check for an answer is non-trivial and could be imprecise.

**Error analysis for text generation tasks** For generation tasks, we consider four types of errors: missing information (required information is not
shown in the output), **contradiction** (the output is contradictory to the input), 3) **hallucination** (the output contains information that cannot be verified by the input), and 4) **ungrammatical**. Figure 3 shows that the proportion of ungrammatical outputs is generally less than 5%. Missing information and contradiction are common errors made by T5, and performance gains generally come from reducing contradiction. Hallucination is not a common error made by T5 except for the highlighted-table-to-text task (ToTTo), where T5 tends to output information of non-highlighted cell values.

**Case study**  We summarize some interesting observations about the model output (more in Appendix H). Compared with T5-base and T5-large, T5-3B’s outputs for text generation tasks tend to be more diverse and creative as shown in Appendix H.2 and H.7. Also, T5-3B sometimes leverages domain knowledge to summarize facts in some tasks such as DART (e.g., describing rating 5 out of 5 as low), while the other two copy the original expressions in the input, as shown in Appendix H.5 and H.6. However, this ability puts T5-3B in the risk of manipulating information and meaning of user request as shown in Appendix H.3.2 and H.4.

### 5 Conclusions

In this paper, we propose the **UNIFIEDSKG** framework to promote systematic research on structured knowledge grounding by unifying 21 SKG tasks. Using **UNIFIEDSKG** as a benchmark, we demonstrate that finetuning T5 on individual tasks achieves state-of-the-art results on almost all 21 tasks. We show that multi-task prefix-tuning benefits most SKG tasks, largely improving the overall performance. For structured knowledge encoding, we find that the effectiveness of encoding variations varies across tasks. Moreover, **UNIFIEDSKG** is a challenging testbed for zero-shot and few-shot learning, shown by the poor results of large PLMs.

### 6 Limitations

**UNIFIEDSKG** establishes a powerful and reproducible starting point for SKG research. New models can be easily applied to diverse SKG tasks, and new tasks can be easily framed based on our standardized abstraction. **UNIFIEDSKG** promotes a systematic study on more general and robust advances in structured knowledge encoding, multi-task learning, zero-shot learning, and few-shot learning for SKG tasks. It also would be interesting to explore general pretraining methods within **UNIFIEDSKG**, which potentially benefit all the unified tasks. When the structured knowledge is too large for GPU memory, we truncate them based on heuristic rules, calling for future study on 1) incorporating retrieval component in SKG, 2) designing sparse attention in T5 for structured knowledge or other means to improve model efficiency.

**UNIFIEDSKG** currently provides the correct type of structured knowledge for each task. However, how a system searches for the correct structured knowledge resources, takes appropriate action, and integrates information and results from multiple structured sources given a user request is still underexplored, which are a prerequisite for building a unified multi-purpose SKG system. Since we select popular tasks from each task family, we risk disproportionality in terms of the data language, domain and population, and we actively welcome diverse, multi-lingual tasks to be added into **UNIFIEDSKG**. Also, the error analysis of SKG can more fine-grained, and we hope our findings can promote future work on systematically studying and decomposing the behavior of PLMs on SKG tasks. Furthermore, training and evaluation data should reflect the intents and linguistic phenomena in the real world (de Vries et al., 2020), suggesting more realistic tasks to be added into **UNIFIEDSKG**.
References


Binyuan Hui, Ruifying Geng, Qiyu Ren, Binhua Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, Pengfei Zhu, and Xiaodan Zhu. 2021. Dynamic hybrid relation network for cross-domain context-dependent semantic parsing.


D. Khashabi, S. Min, T. Khot, A. Sabhwaral, O. Tafjord, P. Clark, and H. Hajishirzi. 2020. Unifieldqa: Crossing format boundaries with a single qa system. EMNLP - findings.


Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariana Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid,


A Contributions

Code implementation Tianbao Xie and Chen Henry Wu implemented the code base of the UNIFIEDSKG framework and experiment pipeline. The code of PICARD and advice from Torsten Scholak sped up the implementation.

Task unification Tianbao Xie, Peng Shi, Michihiro Yasunaga, Chen Henry Wu, and Ming Zhong implemented the 21 tasks into the text-to-text format, adapted the metrics, and verified the performances.

Paper writing Chen Henry Wu and Tianbao Xie finished most part of the paper. Michihiro Yasunaga, Peng Shi, and Chengzu Li added results and analysis for their corresponding parts. Peng Shi drafted related work on SKG with PLMs.

Error analysis and case study Tianbao Xie, Chen Henry Wu, and Michihiro Yasunaga designed and conducted the error analysis for semantic parsing and generation tasks. Authors who participated in the human annotation selected the cases for case study.

Discussion We had three separate weekly meetings, and everyone in the project attended one of them. Torsten Scholak, Ruiqi Zhong, Pengcheng Yin, Victor Zhong, Peng Shi, Rui Zhang, Sida Wang, and Lingpeng Kong actively provided advice. Torsten Scholak provided signals that prefix-tuning would be comparable to fine-tuning. Ruiqi Zhong gave advice on analyzing the effect of model size. Pengcheng Yin and Peng Shi gave advice on analysis on converting structured knowledge into natural language. Pengcheng Yin helped interpret experimental results. Ziyu Yao suggested that we report both sota (w/ extra) and sota (w/o extra) for a fair comparison. Victor Zhong and Bailin Wang gave valuable suggestions on multi-task learning and task transfer analysis. Luke Zettlemoyer, Noah A. Smith, Caiming Xiong, and Dragomir Radev gave valuable comments on research questions and experimental design.

Computing resources We thank Salesforce Research, an Amazon Research Award, ServiceNow Research, and Yale NLP for providing computing resources generously.

Tao Yu designed and led the research.

Acknowledgments

We thank Yifei Min and Libo Qin for their early-stage discussion. We thank Panupong Pasupat and William W. Cohen for their valuable feedback on our initial draft. We thank Qian Liu for his TAPEx code and advice on question answering tasks. We thank wandb for free logging and OpenAI for free Codex usage.
B Results with Full Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>T5-base</th>
<th>T5-large</th>
<th>T5-3B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spider</td>
<td>Match</td>
<td>58.12,,,58.12</td>
<td>66.63,,,66.63</td>
</tr>
<tr>
<td></td>
<td>Exec</td>
<td>60.06,,,60.06</td>
<td>68.28,,,68.28</td>
</tr>
<tr>
<td></td>
<td>Test suite</td>
<td>56.22,,,56.22</td>
<td>64.12,,,64.12</td>
</tr>
<tr>
<td>GraiQA</td>
<td>Match</td>
<td>60.00,,,60.00</td>
<td>67.00,,,67.00</td>
</tr>
<tr>
<td>WebQSP</td>
<td>F1</td>
<td>72.50,,,72.50</td>
<td>73.96,,,73.96</td>
</tr>
<tr>
<td>MTOP</td>
<td>Match</td>
<td>83.89,,,83.89</td>
<td>84.70,,,84.70</td>
</tr>
<tr>
<td></td>
<td>Template</td>
<td>88.85,,,88.85</td>
<td>88.32,,,88.32</td>
</tr>
<tr>
<td>WikiTQ</td>
<td>Acc</td>
<td>36.94,,,36.94</td>
<td>43.30,,,43.30</td>
</tr>
<tr>
<td>WikiSQL</td>
<td>Acc</td>
<td>84.50,,,84.50</td>
<td>86.27,,,86.27</td>
</tr>
<tr>
<td>CompWebQ</td>
<td>Acc</td>
<td>66.71,,,66.71</td>
<td>68.85,,,68.85</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>80.02,,,80.02</td>
<td>81.05,,,81.05</td>
</tr>
<tr>
<td></td>
<td>Hits@1</td>
<td>83.64,,,83.64</td>
<td>85.49,,,85.49</td>
</tr>
<tr>
<td>HybridQA</td>
<td>Acc</td>
<td>54.07,,,54.07</td>
<td>56.95,,,56.95</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>61.85,,,61.85</td>
<td>64.62,,,64.62</td>
</tr>
<tr>
<td>MMQA</td>
<td>Acc</td>
<td>67.29,,,67.29</td>
<td>74.08,,,74.08</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>75.51,,,75.51</td>
<td>81.84,,,81.84</td>
</tr>
<tr>
<td>FeTaQA</td>
<td>BLEU</td>
<td>29.00,,,29.00</td>
<td>30.94,,,30.94</td>
</tr>
<tr>
<td>DART</td>
<td>BLEU</td>
<td>50.62,,,50.62</td>
<td>51.72,,,51.72</td>
</tr>
<tr>
<td>ToTTo</td>
<td>BLEU</td>
<td>48.29,,,48.29</td>
<td>48.95,,,48.95</td>
</tr>
<tr>
<td>MultiWoZ2.1</td>
<td>Joint Acc</td>
<td>57.52,,,57.52</td>
<td>58.23,,,58.23</td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td>20.04,,,20.04</td>
<td>18.84,,,18.84</td>
</tr>
<tr>
<td>SPARQ</td>
<td>Match</td>
<td>50.54,,,50.54</td>
<td>56.69,,,56.69</td>
</tr>
<tr>
<td></td>
<td>Exec</td>
<td>53.95,,,53.95</td>
<td>60.60,,,60.60</td>
</tr>
<tr>
<td></td>
<td>Match (interact)</td>
<td>31.28,,,31.28</td>
<td>37.44,,,37.44</td>
</tr>
<tr>
<td></td>
<td>Exec (interact)</td>
<td>34.36,,,34.36</td>
<td>41.23,,,41.23</td>
</tr>
<tr>
<td>CoSQL</td>
<td>Match</td>
<td>42.30,,,42.30</td>
<td>48.26,,,48.26</td>
</tr>
<tr>
<td></td>
<td>Exec</td>
<td>49.26,,,49.26</td>
<td>56.01,,,56.01</td>
</tr>
<tr>
<td></td>
<td>Match (interact)</td>
<td>12.63,,,12.63</td>
<td>16.72,,,16.72</td>
</tr>
<tr>
<td>SQA</td>
<td>Overall Acc</td>
<td>49.49,,,49.49</td>
<td>59.12,,,59.12</td>
</tr>
<tr>
<td>TabFact</td>
<td>Acc</td>
<td>76.34,,,76.34</td>
<td>81.40,,,81.40</td>
</tr>
<tr>
<td>FEVEROUS</td>
<td>Acc</td>
<td>75.05,,,75.05</td>
<td>79.81,,,79.81</td>
</tr>
<tr>
<td>SQL2Text</td>
<td>BLEC</td>
<td>93.69,,,93.69</td>
<td>93.35,,,93.35</td>
</tr>
<tr>
<td>Logic2Text</td>
<td>BLEC</td>
<td>92.15,,,92.15</td>
<td>92.88,,,92.88</td>
</tr>
</tbody>
</table>

Table 11: Development set performance with full metrics. We do three experiments with different random seeds on representative task of each family and report their averages and standard variances format as $\mu$,$\sigma$.

For the KVRET dataset, instead of the version used in our main tables, we re-run another more widely used pre-processed version (Madotto et al., 2018; Wu et al., 2019; Qin et al., 2020) on T5-base, T5-large and T5-3b. Results are shown in Table 13.

C Input and Output Length Analysis

Linearization of large structured knowledge input (e.g., large tables and KGs) can be arbitrarily long, which needs to be truncated to fit in GPUs with a limited size. The input and output are tokenized by T5Tokenizer in Huggingface’s Transformers.\(^3\) We visualize the length distribution in Figure 5, and details are presented in Table 14. Among the datasets with very long inputs, we choose WikiTableQuestion to study the impact of input length. We visualize the table length distribution and performances with different input truncation lengths in Figure 6. We observe that the accuracy increases as the input becomes longer, motivating future work to study how to effectively encode large structured input, e.g., leveraging sparse attention (Zaheer et al., 2020).

\(^3\)https://huggingface.co/t5-base/tree/main
D Experimental Setup

D.1 Implementation Details

We use T5 (Raffel et al., 2020) as our backbone language model. Each experiment for T5-3B experiments, we use Deepspeed\(^4\) to save memory. We use batch size 32 as default, except WikiTQ,

\(^4\)https://github.com/microsoft/DeepSpeed

WikiSQL, and TabFact, for which for use batch size 128 because we found it to work significantly better. We use the Adafactor optimizer for T5-base and T5-large, and AdamW for T5-3b. We evaluate on the development set for each 500 steps and use the average development set metric for best check-
point selection. For all tasks, we set learning rate to 5e-5 and used linear learning rate decay. All experiments are done on NVIDIA Tesla V100 and NVIDIA Tesla A100.

D.2 Metric Details

For most semantic parsing tasks, we report the exact match accuracy of logical forms, and for task has test suite (Zhong et al., 2020), we add test suite metric to represent model’s performance; an exception is WebQSP, for which we follow previous work to execute the parses and report the F1 score.

For QA, we report the exact match accuracy of answer sets. For data-to-text generation, we report sacre-BLEU (Post, 2018).\(^5\) We use each task’s representative metric used by previous works. For fact verification, we report the accuracy. For high-fidelity NLG, we report BLEC (Shu et al., 2021), which is the exact match between keywords in the formal language and the natural language. Unless specified, we use T5-large and report the development set performance.

\(^5\)Signature: BLEU + case.lc + numrefs.1 + smooth.exp + tok.13a + version.1.4.0

### Table 14: Input and output length for each task’s train set.

<table>
<thead>
<tr>
<th>Task</th>
<th>Structure Input Tokens</th>
<th>Text Input Tokens</th>
<th>Structure Input + Text Input Tokens</th>
<th>Sequence Output Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spider</td>
<td>97.01</td>
<td>1.81</td>
<td>1.17</td>
<td>100.00</td>
</tr>
<tr>
<td>GRAILQA</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>WebQSP</td>
<td>3.40</td>
<td>2.32</td>
<td>94.28</td>
<td>100.00</td>
</tr>
<tr>
<td>MTOP</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>WikiTableQuestions</td>
<td>48.32</td>
<td>27.48</td>
<td>24.18</td>
<td>100.00</td>
</tr>
<tr>
<td>WikiSQL</td>
<td>63.38</td>
<td>25.33</td>
<td>11.29</td>
<td>100.00</td>
</tr>
<tr>
<td>CoreWebQ</td>
<td>1.18</td>
<td>14.52</td>
<td>84.04</td>
<td>100.00</td>
</tr>
<tr>
<td>HybridQA</td>
<td>35.53</td>
<td>50.63</td>
<td>13.8</td>
<td>100.00</td>
</tr>
<tr>
<td>MultiModalQA</td>
<td>63.02</td>
<td>25.67</td>
<td>11.30</td>
<td>100.00</td>
</tr>
<tr>
<td>FeTaQA</td>
<td>60.36</td>
<td>28.62</td>
<td>11.01</td>
<td>100.00</td>
</tr>
<tr>
<td>DART</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>ToTo</td>
<td>95.80</td>
<td>2.87</td>
<td>1.31</td>
<td>100.00</td>
</tr>
<tr>
<td>MultiWoZ</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>98.77</td>
</tr>
<tr>
<td>KVRET</td>
<td>65.08</td>
<td>34.91</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>SPaC</td>
<td>96.70</td>
<td>2.02</td>
<td>1.28</td>
<td>100.00</td>
</tr>
<tr>
<td>CoSQL</td>
<td>96.03</td>
<td>2.23</td>
<td>1.73</td>
<td>100.00</td>
</tr>
<tr>
<td>SQA</td>
<td>64.54</td>
<td>29.74</td>
<td>5.71</td>
<td>100.00</td>
</tr>
<tr>
<td>TabFact</td>
<td>63.22</td>
<td>28.19</td>
<td>8.58</td>
<td>100.00</td>
</tr>
<tr>
<td>FEVEROUS</td>
<td>61.37</td>
<td>22.24</td>
<td>16.39</td>
<td>100.00</td>
</tr>
<tr>
<td>SQL2Text</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Logic2Text</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>Structure Input Tokens</th>
<th>Text Input Tokens</th>
<th>Structure Input + Text Input Tokens</th>
<th>Sequence Output Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spider</td>
<td>97.01</td>
<td>1.81</td>
<td>1.17</td>
<td>100.00</td>
</tr>
<tr>
<td>GRAILQA</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>WebQSP</td>
<td>3.40</td>
<td>2.32</td>
<td>94.28</td>
<td>100.00</td>
</tr>
<tr>
<td>MTOP</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>WikiTableQuestions</td>
<td>48.32</td>
<td>27.48</td>
<td>24.18</td>
<td>100.00</td>
</tr>
<tr>
<td>WikiSQL</td>
<td>63.38</td>
<td>25.33</td>
<td>11.29</td>
<td>100.00</td>
</tr>
<tr>
<td>CoreWebQ</td>
<td>1.18</td>
<td>14.52</td>
<td>84.04</td>
<td>100.00</td>
</tr>
<tr>
<td>HybridQA</td>
<td>35.53</td>
<td>50.63</td>
<td>13.8</td>
<td>100.00</td>
</tr>
<tr>
<td>MultiModalQA</td>
<td>63.02</td>
<td>25.67</td>
<td>11.30</td>
<td>100.00</td>
</tr>
<tr>
<td>FeTaQA</td>
<td>60.36</td>
<td>28.62</td>
<td>11.01</td>
<td>100.00</td>
</tr>
<tr>
<td>DART</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>ToTo</td>
<td>95.80</td>
<td>2.87</td>
<td>1.31</td>
<td>100.00</td>
</tr>
<tr>
<td>MultiWoZ</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>98.77</td>
</tr>
<tr>
<td>KVRET</td>
<td>65.08</td>
<td>34.91</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>SPaC</td>
<td>96.70</td>
<td>2.02</td>
<td>1.28</td>
<td>100.00</td>
</tr>
<tr>
<td>CoSQL</td>
<td>96.03</td>
<td>2.23</td>
<td>1.73</td>
<td>100.00</td>
</tr>
<tr>
<td>SQA</td>
<td>64.54</td>
<td>29.74</td>
<td>5.71</td>
<td>100.00</td>
</tr>
<tr>
<td>TabFact</td>
<td>63.22</td>
<td>28.19</td>
<td>8.58</td>
<td>100.00</td>
</tr>
<tr>
<td>FEVEROUS</td>
<td>61.37</td>
<td>22.24</td>
<td>16.39</td>
<td>100.00</td>
</tr>
<tr>
<td>SQL2Text</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Logic2Text</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

### Table 15: Input and output length for each task’s development set.

<table>
<thead>
<tr>
<th>Task</th>
<th>Structure Input Tokens</th>
<th>Text Input Tokens</th>
<th>Structure Input + Text Input Tokens</th>
<th>Sequence Output Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spider</td>
<td>97.01</td>
<td>1.81</td>
<td>1.17</td>
<td>100.00</td>
</tr>
<tr>
<td>GRAILQA</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>WebQSP</td>
<td>3.40</td>
<td>2.32</td>
<td>94.28</td>
<td>100.00</td>
</tr>
<tr>
<td>MTOP</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>WikiTableQuestions</td>
<td>48.32</td>
<td>27.48</td>
<td>24.18</td>
<td>100.00</td>
</tr>
<tr>
<td>WikiSQL</td>
<td>63.38</td>
<td>25.33</td>
<td>11.29</td>
<td>100.00</td>
</tr>
<tr>
<td>CoreWebQ</td>
<td>1.18</td>
<td>14.52</td>
<td>84.04</td>
<td>100.00</td>
</tr>
<tr>
<td>HybridQA</td>
<td>35.53</td>
<td>50.63</td>
<td>13.8</td>
<td>100.00</td>
</tr>
<tr>
<td>MultiModalQA</td>
<td>63.02</td>
<td>25.67</td>
<td>11.30</td>
<td>100.00</td>
</tr>
<tr>
<td>FeTaQA</td>
<td>60.36</td>
<td>28.62</td>
<td>11.01</td>
<td>100.00</td>
</tr>
<tr>
<td>DART</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>ToTo</td>
<td>95.80</td>
<td>2.87</td>
<td>1.31</td>
<td>100.00</td>
</tr>
<tr>
<td>MultiWoZ</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>98.77</td>
</tr>
<tr>
<td>KVRET</td>
<td>65.08</td>
<td>34.91</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>SPaC</td>
<td>96.70</td>
<td>2.02</td>
<td>1.28</td>
<td>100.00</td>
</tr>
<tr>
<td>CoSQL</td>
<td>96.03</td>
<td>2.23</td>
<td>1.73</td>
<td>100.00</td>
</tr>
<tr>
<td>SQA</td>
<td>64.54</td>
<td>29.74</td>
<td>5.71</td>
<td>100.00</td>
</tr>
<tr>
<td>TabFact</td>
<td>63.22</td>
<td>28.19</td>
<td>8.58</td>
<td>100.00</td>
</tr>
<tr>
<td>FEVEROUS</td>
<td>61.37</td>
<td>22.24</td>
<td>16.39</td>
<td>100.00</td>
</tr>
<tr>
<td>SQL2Text</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Logic2Text</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

For each task in UNIFIEDSKG we search Sanh et al. (2021) for the most similar instructions (if there is no one for use, we create one follow their writing)}
Table 16: Input and output length for each task’s test set.

<table>
<thead>
<tr>
<th>Distribution(%)</th>
<th>Structure Input Tokens</th>
<th>Text Input Tokens</th>
<th>Structure Input + Text Input Tokens</th>
<th>Sequence Output Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="image" /></td>
<td><img src="image2.png" alt="image" /></td>
<td><img src="image3.png" alt="image" /></td>
<td><img src="image4.png" alt="image" /></td>
</tr>
</tbody>
</table>

Figure 6: Length effect on WikiTableQuestion.

D.4 GPT3 and Codex Details

D.4.1 Hyperparameter Settings

Temperature For GPT3 and Codex, we set the decoding temperature to 0 (i.e., greedy decoding without sampling) for Spider, WikiTQ, MultiWoZ and TabFact. We observe a drop of 10% in the exact match metric when set the temperature to 1 by default in OpenAI. For Codex, we tune the temperature from 0 to 1 in a step of 0.1 for DART, SQL2Text, and no significant difference is observed. For GPT3, we do not tune on that to stay within our budget.

Max output length We set max output length to 256 for Spider, WikiTQ, MultiWoZ and SQL2Text, while 4 for TabFact to contain more length in the input side (the concept of max length in GPT3 and Codex is the sum of input tokens length and output tokens length). We set “
” as the stop token.

D.4.2 Prompts We use simple prompt words for each task to concatenate the request, linearized structured knowledge, and context together. For example, for each example in WikiTQ, we format it as “examples

![image](image5.png)  

Paraphrase "[SQL]" to natural language:

D.4 GPT3 and Codex Details

D.4.1 Hyperparameter Settings

Temperature For GPT3 and Codex, we set the decoding temperature to 0 (i.e., greedy decoding without sampling) for Spider, WikiTQ, MultiWoZ and TabFact. We observe a drop of 10% in the exact match metric when set the temperature to 1 by default in OpenAI. For Codex, we tune the temperature from 0 to 1 in a step of 0.1 for DART, SQL2Text, and no significant difference is observed. For GPT3, we do not tune on that to stay within our budget.

Max output length We set max output length to 256 for Spider, WikiTQ, MultiWoZ and SQL2Text, while 4 for TabFact to contain more length in the input side (the concept of max length in GPT3 and Codex is the sum of input tokens length and output tokens length). We set “
” as the stop token.

D.4.2 Prompts We use simple prompt words for each task to concatenate the request, linearized structured knowledge, and context together. For example, for each example in WikiTQ, we format it as “examples

![image](image6.png)  

Paraphrase "[SQL]" to natural language:

D.4 GPT3 and Codex Details

D.4.1 Hyperparameter Settings

Temperature For GPT3 and Codex, we set the decoding temperature to 0 (i.e., greedy decoding without sampling) for Spider, WikiTQ, MultiWoZ and TabFact. We observe a drop of 10% in the exact match metric when set the temperature to 1 by default in OpenAI. For Codex, we tune the temperature from 0 to 1 in a step of 0.1 for DART, SQL2Text, and no significant difference is observed. For GPT3, we do not tune on that to stay within our budget.

Max output length We set max output length to 256 for Spider, WikiTQ, MultiWoZ and SQL2Text, while 4 for TabFact to contain more length in the input side (the concept of max length in GPT3 and Codex is the sum of input tokens length and output tokens length). We set “
” as the stop token.

D.4.2 Prompts We use simple prompt words for each task to concatenate the request, linearized structured knowledge, and context together. For example, for each example in WikiTQ, we format it as “examples

![image](image7.png)  

Paraphrase "[SQL]" to natural language:
age of GPT3 and Codex under the UNIFIED SKG framework is an interesting direction.

D.5 Human Evaluation

Participants of our human evaluation are eight of the authors of this paper. They are familiar with the tasks being evaluated. The human evaluation guideline is shown below.

## General Guideline
1. Each line is a dev set sample, with some inputs (detailed below), a human reference (seq_out) shown in blue, and three model outputs named model1, model2, and model3.
2. Each model output receives a 0-1 score (0 stands for incorrect, and 1 stands for correct). By "correct" we mean "responding to the user request properly and correctly, without grammar or wording mistakes".
3. When an output is incorrect, you specify the type(s) of error, e.g., 1) missing information, 2) contradiction, 3) hallucination, and 4) ungrammatical.

## Task-Specific Details
### DART
1. Task: triples-to-text generation.
2. struct_in: a set of relation-triples joined by `|` and `\`.

### FeTaQA
1. Task: free-form QA
2. question: a question about the table.
3. table: a table represented as a dictionary: {"header": [header item, ...], "rows": [[cell value, ...], ...]}
4. meta: table_page_title | table_section_title

### KVRET
1. Task: dialogue system
2. dialogue: a dialogue represented as a dictionary: {"driver": [request1, ...], "assistant": [response1, ...]}, the last response of the assistant is the human reference.
3. kb: a knowledge base represented as a dictionary: {"header": [header item, ...], "rows": [[cell value, ...], ...]}

### Logic2Text
1. Task: logic expression to text translation
2. table: a table represented as a dictionary: {"caption": table caption, "header": [header item, ...], "rows": [[cell value, ...], ...]}
3. logic_str: logic expression of a statement.

### SQL2Text
1. Task: SQL to text translation
2. query: SQL.

### TOTTo
2. table_page_title and section: table meta information.
3. Visualization of highlighted tables is provided in `\`totto_vis/`.

D.6 Hyperparameters

Shown in Table 17. For semantic parsing tasks, the decoding was done under the greedy search, where we set the beam size to 1 specially. For tasks with a long linearized sequence, we used 1024 as input length to hold the maximum of input; reasons are explained in App. C.

E Training Details

Here we show comparisons of finetuning and prefix-tuning on aspect of training. For prefix-tuning, we use random initialization as done by Li and Liang (2021). In general, prefix-tuning needs more steps than finetuning but has the ability to reach comparable results with continued training.

F Task Unification

F.1 Term Definition

Highlighted tables A highlighted table contains a table, table metadata (such as the title), and a set of highlighted cells which entails the text description (Parikh et al., 2020).

Relation-triples Relation triples are a set of subject-predicate-object triples to capture rich relationships in the data. Many data-to-text tasks such as DART (Nan et al., 2021b) take these relation triples as inputs and generate natural language from them.

Knowledge Graph A knowledge graph is a multi-relational graph composed of entities (nodes) and relations (different types of edges). Each edge is represented as a triple of the form (head entity, relation, tail entity), also called a fact, indicating that two entities are connected by a specific relation (Wang et al., 2017).

Dialogue State and Ontology A dialogue state $s_t$ at any turn $t$ in a dialogue comprises the summary of the dialogue history until turn $t$, such that $s_t$ contains all sufficient information for the system to choose the next action. (Williams et al., 2016) Specifically, it captures the user goals in the conversation in the form of (slot, value) pairs. The set of possible slots is predefined in the ontology $O$, typically domain-dependent, while the values assumed by each slots are provided by the user as a dialogue goal.
<table>
<thead>
<tr>
<th>Task type</th>
<th>Task</th>
<th>Input length</th>
<th>Batch size</th>
<th>Beam size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic Parsing</td>
<td>Spider (Yu et al., 2018)</td>
<td>512</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>GrailQA (Gu et al., 2021)</td>
<td>512</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>WebQSP (Yih et al., 2016)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>MTOP (Li et al., 2021)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>Question Answering</td>
<td>WikiSQL (Zhong et al., 2017)</td>
<td>1024</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>WikiTQ (Pasupat and Liang, 2015)</td>
<td>1024</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>CompWebQ (Talmor and Berant, 2018)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>HybridQA (Chen et al., 2020c)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>MultiModalQA (Talmor et al., 2021)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>FeDiQA (Nan et al., 2021a)</td>
<td>512</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>Data-to-Text</td>
<td>DART (Nan et al., 2021b)</td>
<td>512</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ToTTo (Parikh et al., 2020)</td>
<td>512</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>Conversational</td>
<td>MultiWoZ2.1 (Eric et al., 2019)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>KVRET (Eric et al., 2017)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>SPARQ (Yu et al., 2019b)</td>
<td>512</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CoSQL (Yu et al., 2019a)</td>
<td>512</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SQA (Iyyer et al., 2017)</td>
<td>1024</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>Fact Verification</td>
<td>TabFact (Chen et al., 2020b)</td>
<td>1024</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>FEVEROUS (Aly et al., 2021)</td>
<td>1024</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>High-fidelity NLG</td>
<td>SQL2Text (Shu et al., 2021)</td>
<td>512</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Logic2Text (Chen et al., 2020d)</td>
<td>512</td>
<td>32</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 17: Hyperparameters for each SKG task.

F.2 Linearization

- **Tables.** Following Liu et al. (2021), we linearize the table into a sequence. By inserting several special tokens to indicate the table boundaries, a linearized table can be represented as “col: c_1, ..., c_N row 1: r_1 row 2: r_2... r_M “, N and M are the number of columns and rows.

- **Highlighted tables.** Following Parikh et al. (2020), we represent each highlighted cell by concatenating its value, column headers, and row headers. The table is represented as the concatenation of the page title, section title, and representations of all highlighted cells.

- **Relation-triples and knowledge graphs.** Following Nan et al. (2021b), each relation-triple is linearized as “sub : rela : obj”, and different triples are joined by “|”. The subgraph retrieved from the knowledge graph is treated as a list of relation-triples and we use the same formulation.

- **Ontology.** Following Hosseini-Asl et al. (2020) and Lin et al. (2021), for each slot in ontology, each slot along with all its possible values is formatted as “slot : value_1, ... value_slot_n “, different slot-values are joined by “|”.

F.3 Output Format

When the output is natural language or formal language we do not modify it because it is already in sequence format; a set of answers, we use a comma followed by a space to join the answers; a Boolean value, we map True to “entailed” and False to “refuted”; a dialogue state, we follow Hosseini-Asl et al. (2020) to place its slot-value pairs sequentially.

G Input and Output Examples for Each Task

G.1 Spider

**Structured Input:**

| concert_singer | stadium : stadium_id , location , name , capacity , highest , lowest , average | singer : singer_id , name , country , song_name , song_release_year , age , is_male | concert : concert_id , concert_name , theme , stadium_id , year | singer_in_concert : concert_id , singer_id |

**Request Input:**

How many singers do we have?

**Sequence Output:**

select count(*) from singer

G.2 GRAILQA

**Structured Input:**

soviet red army : m06dr9 | organization. organization_founders government. governmental_body.jurisdiction organization. organization_founder.organizations.founded
Table 18: The comparison of approximate training steps finetuning and prefix-tuning used to reach the decent performance on T5 base. >250000 means we stop the training due to time limitation. Prefix-tuning needs more steps to converge and converges to comparable performances.

<table>
<thead>
<tr>
<th>Task</th>
<th>Finetune</th>
<th>Prefix-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spider</td>
<td>16500</td>
<td>100000</td>
</tr>
<tr>
<td>GraiQA</td>
<td>17000</td>
<td>78000</td>
</tr>
<tr>
<td>WebQSP</td>
<td>15000</td>
<td>8000</td>
</tr>
<tr>
<td>MTOP</td>
<td>30000</td>
<td>60000</td>
</tr>
<tr>
<td>WikiSQL</td>
<td>8500</td>
<td>80000</td>
</tr>
<tr>
<td>WikiTQ</td>
<td>1500</td>
<td>16000</td>
</tr>
<tr>
<td>CompWebQ</td>
<td>3500</td>
<td>27000</td>
</tr>
<tr>
<td>HybridQA</td>
<td>7000</td>
<td>30000</td>
</tr>
<tr>
<td>MultiModalQA</td>
<td>6000</td>
<td>40000</td>
</tr>
<tr>
<td>FctTQQA</td>
<td>11000</td>
<td>20000</td>
</tr>
<tr>
<td>DART</td>
<td>7000</td>
<td>250000</td>
</tr>
<tr>
<td>ToTTo</td>
<td>12000</td>
<td>&gt;250000</td>
</tr>
<tr>
<td>MultiWoZ2.1</td>
<td>6000</td>
<td>40000</td>
</tr>
<tr>
<td>KVRET</td>
<td>4000</td>
<td>40000</td>
</tr>
<tr>
<td>SPARc</td>
<td>2000</td>
<td>6400</td>
</tr>
<tr>
<td>CoSQL</td>
<td>38000</td>
<td>100000</td>
</tr>
<tr>
<td>SQA</td>
<td>27000</td>
<td>&gt;250000</td>
</tr>
<tr>
<td>TabFact</td>
<td>8000</td>
<td>210000</td>
</tr>
<tr>
<td>FEVEROUS</td>
<td>12000</td>
<td>40000</td>
</tr>
<tr>
<td>SQL2Text</td>
<td>3000</td>
<td>10000</td>
</tr>
<tr>
<td>Logic2Text</td>
<td>3500</td>
<td>10000</td>
</tr>
</tbody>
</table>

The person who founded the Soviet Red Army also founded what government agency?

**Sequence Output:**
(Cut Bank)

**G.4 WebQsp**

**Structured Input:**
Spain: m.06mkj | m.06mkj location.location. contains m.0g3qgy | m.06mkj location.location. contains m.02qf5mh | m.0j5_3sv government. government_position_held. office_position_or_title m.0j5_3sz | m.06mkj location.location. contains m.02zb43k | (omitted to save space) | m.06mkj government. government. government_location. contains m.09k5hy | m.010wswjc government. government_position_held. office_position_or_title m.0j5_3sz | m.06mkj location.location. contains m.02z98t5 | m.06mkj location.location. contains m.03qcr00

**Request Input:**
What movie was produced by Dan Cohen and features Liam Hemsworth as an actor?

**Sequence Output:**
Cut Bank

**G.5 MTOP**

**Structured Input:**
IN:GET: MESSAGE, WEATHER, ALARM, INFO_RECIPES, STORIES_NEWS, REMINDER, RECIPES, EVENT, CALL_TIME, LIFE_EVENT, INFO_CONTACT, CONTACT, TIMER, REMINDER_DATE_TIME, AGE, SUNRISE, EMPLOYER, EDUCATION_TIME, JOB, AVAILABILITY, (omitted to save space)

(omitted to save space)"
Request Input:
call Nicholas and Natasha

Sequence Output:
[IN:CREATE_CALL [SL:CONTACT Nicholas ] [SL:CONTACT Natasha ] ]

G.6 WikiSQL
Structured Input:

Request Input:
what position does the player who played for butler cc (ks) play?

Sequence Output:
guard-forward

G.7 WikiTableQuestions
Structured Input:

Request Input:
which team won previous to crettyard?

Sequence Output:
wolfe tones

G.8 HybridQA
Structured Input:
col : position | athlete | nationality | time row 1 : 1 | patrick makau musyoki | kenya | 2:03.38 row 2 : 2 | stephen kwelio chemlany | kenya | 2:07.55 row 3 : 3 | edwin kimaiyo | kenya | 2:09.50 row 4 : 4 | felix limo | kenya | 2:10.38 row 5 : 5 | scott overall | united kingdom | 2:10.55 row 6 : 6 | ricardo serrano | spain | 2:13.32 row 7 : 7 | pedro nimo | spain | 2:13.34 row 8 : 8 | simon munyutu | france | 2:14.20 row 9 : 9 | driss el himer | france | 2:14.46 row 10 : 10 | hendrik ramaala | south africa | 2:16.00 passages: ricardo serrano (athlete): at the 2011 iaaf world cross country championships he was 89th overall. his marathon debut followed later that year and he was sixth at the 2011 berlin marathon with a time of 2:13.32 hours. | spain: with an area of 505,990 km2 ( 195,360 sq mi ), spain is the largest country in southern europe, the second largest country in western europe and the european union, and the fourth largest country in the european continent. by population (about 47 million), spain is the sixth largest in europe and the fifth in the european union.

Request Input:
what place was achieved by the person who finished the berlin marathon in 2:13.32 in 2011 the first time he competed in a marathon?

Sequence Output:
sixth

G.9 MultiModalQA
Structured Input:

Request Input:
for which film did ben piazza play the role of mr. simms?

Sequence Output:
mask

G.10 FeTaQA
Structured Input:
andy karl | awards and nominations col : year | award | category | work | result row 1 : 2013 | drama desk award | outstanding featured actor in a musical | the mystery of edwin drood | nominated row 2 : 2013 | broadway.com audience awards | favorite onstage pair (with jessie mueller) | the mystery of edwin drood | nominated row 3 : 2014 |
when did andy karl win the olivier award and for which of his work?

for his performance in groundhog day, andy karl received the 2017 olivier award for best actor in a musical.

Daniel Henry Chamberlain was the 76th Governor of South Carolina from 1874.

i am looking to book a train that is leaving from cambridge to bishops stortford on friday.

booking was successful. reference number is: bmkptg6. can i help you with anything else today?

i can help you with that. what is your price range?

i need to book a hotel in the east that has 4 stars.

Daniel Henry Chamberlain was the 76th Governor of South Carolina from 1874.
col : poi | poi_type | address | distance |
traffic_info row 1 : sigona farmers market |
grocery store | 638 amherst st | 3 miles | car
collision nearby row 2 : cafe venetia | coffee
or tea place | 269 alger dr | 1 miles | car
collision nearby row 3 : 5672 barringer street |
certain address | 5672 barringer street | 5 miles | no traffic row 4 : valero | gas station
| 200 alester ave | 2 miles | road block nearby row 5 : stanford childrens health | hospital
| 899 ames ct | 5 miles | moderate traffic row 6 : palo alto garage r | parking garage | 481
amaranta ave | 1 miles | moderate traffic row 7 : teavana | coffee or tea place | 145 amherst st
| 1 miles | road block nearby row 8 : willows
market | grocery store | 409 bollard st | 5 miles | no traffic

**Context:**
How many car models are produced by each maker?
List the count and the maker full name.

**Sequence Output:**
select t1.model from model_list as t1 join car_makers as t2 on t1.maker = t2.id group by t2.id order by count ( * ) desc limit 1

**G.17 SQA**

**Structured Input:**
col : date | number | type | captain |
casualties | sunk by row 1 : 4 may 1943 | u-209
| viic | heinrich brodda | 46 | catalina flying-boat of no. 5 squadron row 2 : 5 may 1943 | u-638
| viic | oskar staudinger | 44 | hms sunflower row 3 : 5 may 1943 | u-531
| ixc/40 | herbert neckel | 54 | hms vidette row 4 : 6 may 1943 | u-192
| ixc/40 | ulrich folkers | 54 | hms ombi | hms snowflake row 6 : 6 may 1943 | u-630
| viic | werner winkler | 47 | hms vidette row 7 : 6 may 1943 | u-438
| viic | heinrich hensohn | 48 | hms pelican

**Request Input:**
which captain was not oskar staudinger?

**Context:**
who were the captains of those boats? | what boats were lost on may 5?

**Sequence Output:**
herbert neckel

**G.18 TabFact**

**Structured Input:**
col : round | clubs remaining | clubs involved | winners from previous round | new entries this round | leagues entering at this round row 1 : first round | 156 | 86 | none | 86 | tff third league & turkish regional amateur league row 2 : second round | 113 | 108 | 43 | 65 | superlig & tff first league & tff second league row 3 : third round | 99 | 54 | 54 | none | none row 4 : fourth round | 32 | 32 | 27 | 5 | su0fcp row 5 : fifth round | 5 | 54 | none | none | none row 6 : group stage | 8 | 8 | 8 | 8 | none | none row 7 : semi-finals | 4 | 4 | 4 | none | none row 8 : finals | 2 | 2 | 2 | none | none

**Request Input:**
during the third round of the turkish cup , there be no new entry during that stage

**Sequence Output:**
entailed
G.19 FEVEROUS

Structured Input:

<table>
<thead>
<tr>
<th>col</th>
<th>title</th>
<th>narrator</th>
<th>aired between</th>
</tr>
</thead>
<tbody>
<tr>
<td>original air date</td>
<td>us viewers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>magic is coming</td>
<td>giancarlo esposito</td>
<td>&quot;a land without magic&quot;</td>
<td>september 30, 2012 (2012-09-30)</td>
</tr>
<tr>
<td>lacey</td>
<td>april 14, 2013 (2013-04-14)</td>
<td>5.17 row 3 : 3</td>
<td>&quot;journey to neverland&quot;</td>
</tr>
<tr>
<td>&quot;dark swan rises: a once upon a time fan</td>
<td>howard parker</td>
<td>&quot;operation mongoose, part 2&quot; &quot; the</td>
<td>september 27, 2015 (2015-09-27)</td>
</tr>
</tbody>
</table>

Request Input:

list of once upon a time (debuted october 23, 2011) episodes consists of 10 specials, one of the narrators alan dale.

Sequence Output:

refuted

G.20 SQL2Text

Structured Input:

None

Request Input:

SELECT Fname FROM FACULTY WHERE Rank = "Professor" ORDER BY Fname

Sequence Output:

What are the first names for all faculty professors, ordered by first name?

T5-base prediction (incorrect):

select name, capacity from stadium except select t2.name from concert as t1 join stadium as t2 on t1.stadium_id = t2.stadium_id where t1.year = 2014

T5-large prediction (correct):

select name, capacity from stadium except select t2.name from concert as t1 join stadium as t2 on t1.stadium_id = t2.stadium_id where t1.year = 2014

T5-3B prediction (correct):

select name, capacity from stadium except select t2.name from concert as t1 join stadium as t2 on t1.stadium_id = t2.stadium_id where t1.year = 2014

H Case study

H.1 Spider case

H.1.1 Case 1:

Structured Input:

| concert_singer | stadium : stadium_id , location , name , capacity , highest , lowest , average | singer : singer_id , name , country , song_name , song_release_year , age , is_male | concert : concert_id , concert_name , theme , stadium_id , year | singer_in_concert : concert_id , singer_id |

Request Input:

What are the names of all stadiums that did not have a concert in 2014?

Sequence Output:

select name from stadium except select t2.name from concert as t1 join stadium as t2 on t1.stadium_id = t2.stadium_id where t1.year = 2014

T5-base prediction (incorrect):

select name from stadium except select stadium_name from concert where year = 2014

T5-large prediction (correct):

select name from stadium except select t2.name from concert as t1 join stadium as t2 on t1.stadium_id = t2.stadium_id where t1.year = 2014

T5-3B prediction (correct):

select name from stadium except select t2.name from concert as t1 join stadium as t2 on t1.stadium_id = t2.stadium_id where t1.year = 2014

H.1.2 Case 2:

Structured Input:

| concert_singer | stadium : stadium_id , location , name , capacity , highest , lowest , average | singer : singer_id , name , country , song_name , song_release_year , age , is_male | concert : concert_id , concert_name , theme , stadium_id , year | singer_in_concert : concert_id , singer_id |

Request Input:

What is the name and capacity for the stadium with highest average attendance?

Sequence Output:

select name, capacity from stadium order by average desc limit 1

T5-base prediction (incorrect):

select name, capacity from stadium order by avg(amount) desc limit 1

T5-large prediction (correct):

select name, capacity from stadium order by average desc limit 1
T5-3B prediction (correct):

```sql
select name, capacity from stadium order by average desc limit 1
```

H.1.3 Case 3:
Structured Input:
```
| pets_1 | student : stuid, lname, fname, age, sex, major, advisor, city_code | has_pet : stuid, petid | pets : petid, pettype (cat, dog), pet_age, weight
```

Request Input:
Find the first name of students who have cat or dog pet.

Sequence Output:
```
select distinct t1.fname from student as t1 join has_pet as t2 on t1.stuid = t2.stuid where t2.pettype = 'cat' or t2.pettype = 'dog'
```

T5-base prediction (incorrect):
```
select t1.fname from student as t1 join has_pet as t2 on t1.stuid = t2.stuid where t2.pettype = "cat" or t2.pettype = "dog"
```

T5-large prediction (incorrect):
```
select t1.fname from student as t1 join has_pet as t2 on t1.stuid = t2.stuid where t2.pettype = "cat" or t2.pettype = "dog"
```

T5-3B prediction (correct):
```
select t1.fname from student as t1 join has_pet as t2 on t1.stuid = t2.stuid join pets as t3 on t2.petid = t3.petid where t3.pettype = "cat" or t3.pettype = "dog"
```

H.2 FeTaQA case
Structured Input:
```
te aroha (new zealand electorate) | 1890 election col : party | party | candidate | votes % | [non utf-8 token] row 1 : - | - | independent | william shepherd allen | 786 | 56.34 | - row 2 : - | liberal | william fraser | 609 | 43.65 | - row 3 : majority | majority | majority | 175 | 12.54 | - row 4 : turnout | turnout | turnout | 1,395 | 48.60 | - row 5 : registered electors | registered electors | registered electors | 2,870 | - | -
```

Request Input:
what were the voting results in the te aroha election by the candidate?

Sequence Output:
```
votes for allen and fraser were 786 and 609 respectively.
```

T5-base prediction (correct):
```
votes for allen and fraser were 786 and 609 respectively.
```

T5-large prediction (incorrect):
```
it will not rain in alhambra this weekend.
```

T5-3B prediction (incorrect):
```
there will be no rain this weekend in alhambra.
```

H.3 KVRET case
H.3.1 Case 1:
Structured Input:
```
col : location | monday | tuesday | wednesday | thursday | friday | saturday | sunday | today row 1 : danville | clear skies, low of 30f, high of 50f | cloudy, low of 20f, high of 40f | raining, low of 80f, high of 100f | dew, low of 80f, high of 90f | clear skies, low of 70f, high of 80f | frost, low of 40f, high of 60f | overcast, low of 20f, high of 30f | monday row 2 : - | - | - | - | - | - | - | - row 3 : alhambra | windy, low of 90f, high of 100f | hail, low of 90f, high of 100f | clear skies, low of 80f, high of 90f | rain, low of 70f, high of 80f | stormy, low of 70f, high of 80f | rain, low of 60f, high of 70f | hail, low of 40f, high of 60f | monday ...
```

Request Input:
alhambra please.

Context Input:
what city are you inquiring about? | is it going to be raining this weekend

Sequence Output:
it will be raining on saturday and hailing on sunday in alhambra.

T5-base prediction (correct):
it will be raining on saturday and hailing on sunday in alhambra.

T5-large prediction (incorrect):
```
it will not rain in alhambra this weekend.
```

T5-3B prediction (incorrect):
```
there will be no rain this weekend in alhambra.
```

H.3.2 Case 2:
Structured Input:
```
col : event | time | date | room | agenda | party row 1 : tennis activity | 5pm | wednesday | - | - | mother row 2 : taking medicine | 9am | the 3rd | - | - | sister row 3 : swimming activity | 11am | friday | - | - | father row 4 : conference | 6pm | the 8th | 100 conference room | go over budget | infrastructure team row 5 : meeting | 4pm | the 10th | conference room 102 | go over quarterly report | infrastructure team row 6 : dentist appointment | 10am | the 13 th | - | - | jeff row 7 : dentist appointment | 11am | the 5th | - | - | jeff row 8 : lab appointment | 2pm | the 4th | - | - | sister
```

T5-large prediction (incorrect):
in the te aroha election, allen won with a majority of 175 votes.

T5-3B prediction (correct):
in the 1890 te aroha election, the result was a victory for the independent candidate, william shepherd allen, who received 786 votes against 609 for the liberal candidate, william fraser.

H.3 KVRET case
H.3.1 Case 1:
Structured Input:
```
col : location | monday | tuesday | wednesday | thursday | friday | saturday | sunday | today row 1 : danville | clear skies, low of 30f, high of 50f | cloudy, low of 20f, high of 40f | raining, low of 80f, high of 100f | dew, low of 80f, high of 90f | clear skies, low of 70f, high of 80f | frost, low of 40f, high of 60f | overcast, low of 20f, high of 30f | monday row 2 : - | - | - | - | - | - | - | - row 3 : alhambra | windy, low of 90f, high of 100f | hail, low of 90f, high of 100f | clear skies, low of 80f, high of 90f | rain, low of 70f, high of 80f | stormy, low of 70f, high of 80f | rain, low of 60f, high of 70f | hail, low of 40f, high of 60f | monday ...
```

Request Input:
alhambra please.

Context Input:
what city are you inquiring about? | is it going to be raining this weekend

Sequence Output:
it will be raining on saturday and hailing on sunday in alhambra.

T5-base prediction (correct):
it will be raining on saturday and hailing on sunday in alhambra.

T5-large prediction (incorrect):
it will not rain in alhambra this weekend.

T5-3B prediction (incorrect):
```
there will be no rain this weekend in alhambra.
```

H.3.2 Case 2:
Structured Input:
```
col : event | time | date | room | agenda | party row 1 : tennis activity | 5pm | wednesday | - | - | mother row 2 : taking medicine | 9am | the 3rd | - | - | sister row 3 : swimming activity | 11am | friday | - | - | father row 4 : conference | 6pm | the 8th | 100 conference room | go over budget | infrastructure team row 5 : meeting | 4pm | the 10th | conference room 102 | go over quarterly report | infrastructure team row 6 : dentist appointment | 10am | the 13 th | - | - | jeff row 7 : dentist appointment | 11am | the 5th | - | - | jeff row 8 : lab appointment | 2pm | the 4th | - | - | sister
```

T5-3B prediction (correct):
in the 1890 te aroha election, the result was a victory for the independent candidate, william shepherd allen, who received 786 votes against 609 for the liberal candidate, william fraser.
Request Input:
when is the date and time of my
dentist appointment?

Context Input:
None

Sequence Output:
your dentist appointment is at 10am on the_13th.

T5-base prediction (correct):
you have two dentist appointments scheduled. one
on the_5th at 11am and another on the_13th at
10am.

T5-large prediction (correct):
your dentist appointment is on the_13th at 10 am.

T5-3B prediction (incorrect):
you have three dentist appointments scheduled.
one on the_5th at 11am, another on the_13th at
10am, and the last is on the_5th at 11am.

H.4 SQL2Text case
Query:
SELECT company , main_industry FROM company
WHERE headquarters != 'USA'

Sequence Out:
Show the company name and the main industry for
all companies whose headquarters are not from
USA.

T5-base prediction (incorrect):
What are the companies and main industries of
companies whose headquarters are not "USA"?

T5-large prediction (correct):
What are the companies and main industries of
the companies whose headquarters are not located
in the United States?

T5-3B prediction (incorrect):
What are the main industries of companies whose
headquarters are not in the USA?

H.5 DART case
Structured Input:

Sequence Output:
The Cotto is a highly rated English coffee shop
in the city centre near The Portland Arms, and
prices range more than [non utf-8 token]30.

T5-base prediction (correct):
Cotto is a coffee shop providing English food in
the more than [non utf-8 token]30 price range.
It is located in the city centre. It is near The
Portland Arms. Its customer rating is high.

T5-large prediction (correct):
Cotto is a coffee shop providing English food in
the more than [non utf-8 token]30 price range.
It is located in the city centre. It is near The
Portland Arms. Its customer rating is high.

T5-3B prediction (correct):
Cotto is a coffee shop located in the city
centre near The Portland Arms. It serves English
food with a price range of more than [non utf-8
token]30 and has a high customer rating.

H.6 Logic2Text case
Structured Input:
Caption: 2008 san diego chargers season | week,
date, time, opponent, result, game site, nfl
recap, record
Request Input:
eq { count { filter_eq { filter_eq { all_rows ;
game site ; qualcomm stadium } ; time ; 5:15 pm } } ; 3 } = true

Sequence Output:
in the 2008 san diego chargers season, among
the games that were played in qualcomm stadium ,
3 of them started at 5:15 pm .

T5-base prediction (incorrect):
in the 2008 san diego chargers season, when the
game was at qualcomm stadium, there were three
times the time was 5:15 pm.

T5-large prediction (incorrect):
in the 2008 san diego chargers season, when the
game was at qualcomm stadium, there were 3 times
the time was 5:15 pm.

T5-3B prediction (correct):
in the 2008 san diego chargers season, among the
games played at qualcomm stadium, 3 of them
started at 5:15 pm.

H.7 ToTTo case
Structured Input: See Figure 7.

Sequence Output:
Alisson Perticheto placed 18th at the 2013
Junior Worlds, 17th at the 2014 Four Continents
and 16th at the 2015 Four Continents.

T5-base prediction (incorrect):
Alisson Perticheto finished 18th at the Junior
Worlds and 17th at the Four Continents.
T5-large prediction (incorrect):
Alisson Perticheto placed 17th at the 2014 Four Continents and 16th at the 2015 Junior Worlds.

T5-3B prediction (correct):
Alisson Perticheto finished 17th at the 2014 Four Continents, 16th at the 2015 Four Continents, and 18th at the 2013 Junior Worlds.

I. Natural Language Template Examples

I.1 Spider Template

Overall Description Template:
{db id} contains tables such as {table1 name}, {table2 name}

Primary Key Template:
{primary key} is the primary key.

Table Description Template:
Table {table name} has column such as {column 1 name}, {column 2 name}, ...

Foreign Keys Description Template:
The {column1 name} of {table 1} is the foreign key of {column2 name} of {table 2}

I.2 TabFact Template

Template Examples:
Table 1-14978398-2:
The {version} of song Comme j'ai mal has a length of {length} in album {album} remixed by {remixed by} in year {year}

Table 1-15187735-12:
On {date} in 1936 VFL Season, the home team {home team} and away team {away team} had a game at venue {venue} with a crowd of {crowd}; the home team score is {home team score} and the away team score is {away team score}

I.3 WikiSQL Template

Template Example:
Table 1-14240688-1:
in {year} were in division {division}, {league} ranked {regular season}, made it to {playoffs} of the playoffs, made it to <{open cup}> in the open cup, and kept an average attendance of {avg attendance}

Table 2-12997882-1:
On {date} in 2008 European Figure Skating, the home team {home team} and away team {away team} had a game at venue {venue} with a crowd of {crowd}; the home team score is {home team score} and the away team score is {away team score}

Table 1-13740746-1:
Episode number {ep no} of gerry anderson's new captain scarlet with a title of {title} is directed by {director} and written by {written by}; its original air date is {original air date}; the production number is {production no}