TabGenie: A Toolkit for Table-to-Text Generation

Zdeněk Kasner[1] **Ekaterina Garanina**[1,2] **Ondřej Plátek**[1] **Ondřej Dušek**[1]

[1]Charles University, Czechia

[2]University of Groningen, The Netherlands

{kasner,oplatek,odusek}@ufal.mff.cuni.cz

e.garanina@student.rug.nl

Abstract

Heterogenity of data-to-text generation datasets limits the research on data-to-text generation systems. We present TabGenie – a toolkit which enables researchers to explore, preprocess, and analyze a variety of data-to-text generation datasets through the unified framework of *table-to-text generation*. In TabGenie, all inputs are represented as tables with associated metadata. The tables can be explored through a web interface, which also provides an interactive mode for debugging table-to-text generation, facilitates side-by-side comparison of generated system outputs, and allows easy exports for manual analysis. Furthermore, TabGenie is equipped with command line processing tools and Python bindings for unified dataset loading and processing. We release TabGenie as a PyPI package[1] and provide its open-source code and a live demo at https://github.com/kasnerz/tabgenie.

1 Introduction

Building and evaluating data-to-text (D2T) generation systems (Gatt and Krahmer, 2018; Sharma et al., 2022) requires understanding the data and observing system behavior. It is, however, not trivial to interact with the large volume of D2T generation datasets that have emerged in the last years (see Table 1). Although research on D2T generation benefits from platforms providing unified interfaces, such as HuggingFace Datasets (Lhoest et al., 2021) or the GEM benchmark (Gehrmann et al., 2021), these platforms still leave the majority of the data processing load on the user.

A key component missing from current D2T tools is the possibility to visualize the input data and generated outputs. Visualization plays an important role in examining and evaluating scientific data (Kehrer and Hauser, 2013) and can help D2T generation researchers to make more informed design choices. A suitable interface can also encourage researchers to step away from unreliable automatic metrics (Gehrmann et al., 2022) and focus on manual error analysis (van Miltenburg et al., 2021, 2023).

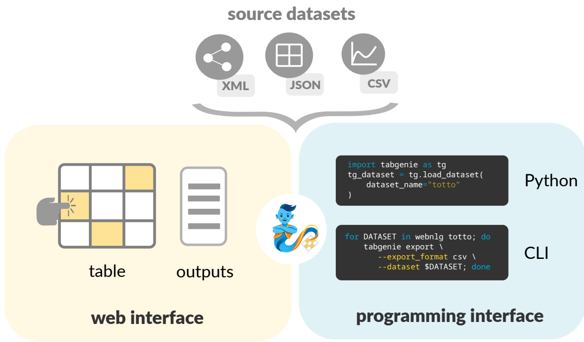

Figure 1: TabGenie provides a way to handle various data-to-text generation datasets through a unified web and programming interface. The *web interface* enables interactive exploration and analysis of datasets and model outputs, while the *programming interface* provides unified data loaders and structures.

Along with that, demands for a *unified input data format* have recently been raised with multi-task training for large language models (LLMs) (Sanh et al., 2022; Scao et al., 2022; Ouyang et al., 2022, *inter alia*). Some works have used simple data linearization techniques for converting structured data to a textual format, in order to align it with the format used for other tasks (Xie et al., 2022; Tang et al., 2022). However, linearizations are using custom preprocessing code, leading to discrepancies between individual works.

In this paper, we present TabGenie – a multi-purpose toolkit for interacting with D2T generation datasets and systems designed to fill these gaps. On a high level, the toolkit consists of (a) an interactive web interface, (b) a set of command-line processing tools, and (c) a set of Python bindings (see Figure 1).

The cornerstone of TabGenie is a **unified data**

[1]https://pypi.org/project/tabgenie/

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 444–455
July 10-12, 2023 ©2023 Association for Computational Linguistics

Dataset	Source	Data Type	Number of examples train	dev	test	License
CACAPO	van der Lee et al. (2020)	Key-value	15,290	1,831	3,028	CC BY
DART†	Nan et al. (2021)	Graph	62,659	2,768	5,097	MIT
E2E†	Dusek et al. (2019)	Key-value	33,525	1,484	1,847	CC BY-SA
EventNarrative	Colas et al. (2021)	Graph	179,544	22,442	22,442	CC BY
HiTab	Cheng et al. (2022)	Table w/hl	7,417	1,671	1,584	C-UDA
Chart-To-Text	Kantharaj et al. (2022)	Chart	24,368	5,221	5,222	GNU GPL
Logic2Text	Chen et al. (2020b)	Table w/hl + Logic	8,566	1,095	1,092	MIT
LogicNLG	Chen et al. (2020a)	Table	28,450	4,260	4,305	MIT
NumericNLG	Suadaa et al. (2021)	Table	1,084	136	135	CC BY-SA
SciGen	Moosavi et al. (2021)	Table	13,607	3,452	492	CC BY-NC-SA
SportSett:Basketball†	Thomson et al. (2020)	Table	3,690	1,230	1,230	MIT
ToTTo†	Parikh et al. (2020)	Table w/hl	121,153	7,700	7,700	CC BY-SA
WebNLG†	Ferreira et al. (2020)	Graph	35,425	1,666	1,778	CC BY-NC
WikiBio†	Lebret et al. (2016)	Key-value	582,659	72,831	72,831	CC BY-SA
WikiSQL†	Zhong et al. (2017)	Table + SQL	56,355	8,421	15,878	BSD
WikiTableText	Bao et al. (2018)	Key-value	10,000	1,318	2,000	CC BY

Table 1: The list of datasets included in TabGenie. Glossary of data types: *Key-value*: key-value pairs, *Graph*: subject-predicate-object triples, *Table*: tabular data (*w/hl*: with highlighted cells), *Chart*: chart data, *Logic / SQL*: strings with logical expressions / SQL queries. The datasets marked with † were already present on Huggingface Datasets. We uploaded the rest of the datasets to our namespace: `https://huggingface.co/kasnerz`.

representation. Each input represented is as a matrix of m columns and n rows consisting of individual cells accompanied with metadata (see §2). Building upon this representation, TabGenie then provides multiple features for unified workflows with table-to-text datasets, including:

1. visualizing individual dataset examples in the tabular format (§3.1),
2. interacting with table-to-text generation systems in real-time (§3.2),
3. comparing generated system outputs (§3.2),
4. loading and preprocessing data for downstream tasks (§4.1),
5. exporting examples and generating spreadsheets for manual error analysis (§4.2).

In §6, we present examples of practical use-cases of TabGenie in D2T generation research.

2 Data

We currently include 16 datasets listed in Table 1 in TabGenie, covering many subtasks of D2T generation. All the datasets are available under a permissive open-source license.

2.1 Data Format

The inputs in D2T generation datasets may not consist only of tables, but also of e.g. graphs or key-value pairs. However, we noticed that in many cases, converting these formats to tables requires only minimal changes to the data structure while allowing a unified data representation and visualization. This conversion narrows down the task of D2T generation as the task of generating description for a tabular data, i.e. table-to-text generation (Parikh et al., 2020; Liu et al., 2022; Gong et al., 2020).

In our definition, a *table* is a two-dimensional matrix with m columns and n rows, which together define a grid of $m \times n$ cells. Each cell contains a (possibly empty) text string. A continuous sequence of cells $\{c_i, \ldots, c_{i+k}\}$ from the same row or column may be merged, in which case the values of $\{c_{i+1}, \ldots, c_{i+k}\}$ are linked to the value of c_i. A cell may be optionally marked as a *heading*, which is represented as an additional property of the cell.[2] To better accommodate the format of datasets such as ToTTo (Parikh et al., 2020) or HiTab (Cheng et al., 2022), we also allow individual cells to be *highlighted*. Highlighted cells are assumed to be preselected for generating the output description.

The tables may be accompanied with an additional set of properties (see Figure 2) – an example of such a property is a *"title"* of the table in WikiBio (Lebret et al., 2016) or a *"category"* in WebNLG (Gardent et al., 2017). We represent prop-

[2]The headings are typically located in the first row or column, but may also span multiple rows or columns and may not be adjacent.

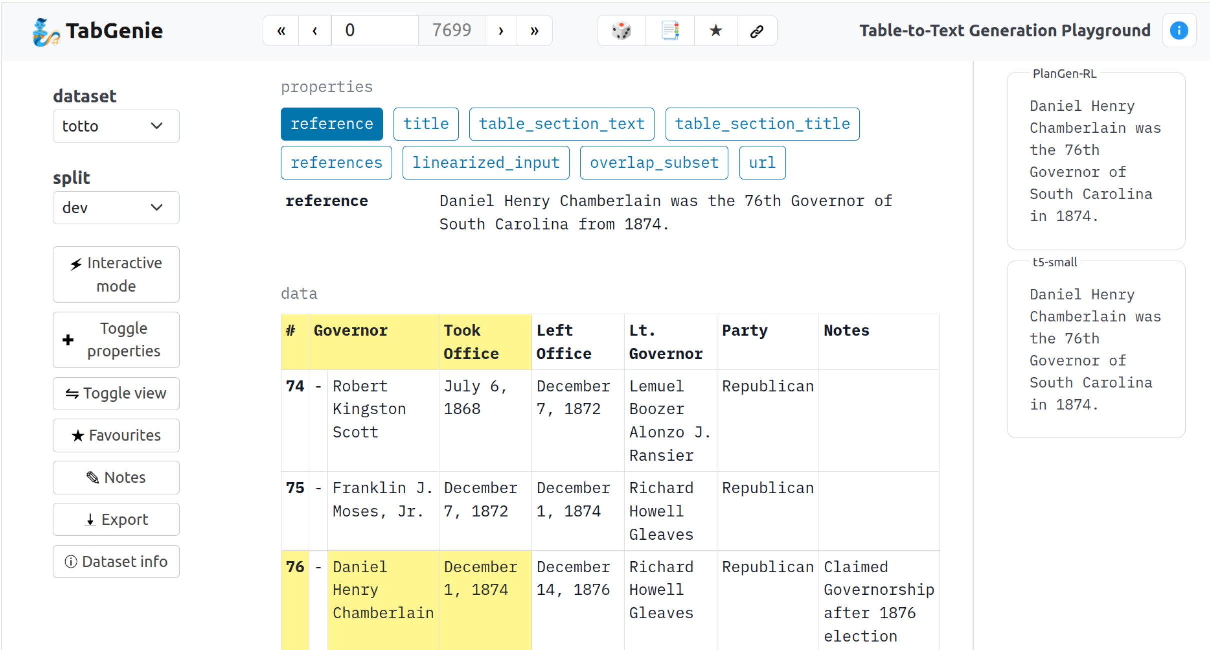

Figure 2: The web interface of TabGenie. The **left panel** and the **navigation bar** contains user controls; the **center panel** shows table properties and table content; the **right panel** contains system outputs.

erties as key-value pairs alongside the table. The properties may be used for generating the table description.

2.2 Data Transformation

We aim to present the data as true to the original format as possible and only make some minor changes for datasets which do not immediately adhere to the tabular format:

- For graph-to-text datasets, we format each triple as a row, using three columns labeled *subject*, *predicate*, and *object*.
- For key-value datasets, we use two columns with keys in the first column as row headings.
- For SportSett:Basketball (Thomson et al., 2020), we merge the *box score* and *line score* tables and add appropriate headings where necessary.

Moreover, for ToTTo (Parikh et al., 2020), we also provide our custom, improved header cells highlighting (details are given in Appendix A).

2.3 Data Loading

To ease the data distribution, we load all the datasets using the Huggingface `datasets` package (Lhoest et al., 2021), which comes equipped with a data downloader. Out of 16 datasets we are using, 7 were already available in Huggingface datasets, either through the GEM benchmark (Gehrmann et al., 2021) or other sources. We publicly added the 9 remaining datasets (see Table 1).

TabGenie also supports adding custom data loaders. Creating a data loader consists of simple sub-classing the data loader class and overriding a single method for processing individual entries, allowing anyone to add their custom dataset.

3 Web Interface

TabGenie offers a user-friendly way to interact with table-to-text generation datasets through the *web interface*. The interface can be rendered using a local server (cf. §4.2) and can be viewed in any modern web browser. The interface features a simple, single-page layout, which contains a navigation bar and three panels containing user controls, input data, and system outputs (see Figure 2). Although the interface primarily aims at researchers, it can be also used by non-expert users.

3.1 Content Exploration

TabGenie renders input data as HTML tables. This approach provides superior visualizations to existing data viewers, especially in the case of large and hierarchical tables.[3]

[3]Compare, e.g., with the ToTTo dataset in Huggingface Datasets for which the table is provided in a single field called *"table"*: `https://huggingface.co/datasets/totto`

In the web interface, users can navigate through individual examples in the dataset sequentially, access an example using its index, or go to a random example. The users can add notes to examples and mark examples as favorites for accessing them later. The interface also shows the information about the dataset (such as its description, version, homepage, and license) and provides an option to export the individual examples (see §4.2).

3.2 Interactive Mode

TabGenie offers an *interactive mode* for generating an output for a particular example on-the-fly. The user can highlight different cells, edit cell contents, and edit parameters of the downstream processor. For example, the user can prompt a LLM for table-to-text generation and observe how it behaves while changing the prompt.

The contents of a table are processed by a processing *pipeline*. This pipeline takes table contents and properties as input, processes them with a sequence of modules, and outputs HTML code. The modules are custom Python programs which may be re-used across the pipelines.

TabGenie currently provides two basic pipelines: (1) calling a generative language model through an API with a custom prompt, and (2) generating graph visualizations of RDF triples. We describe a case-study for the model API pipeline in §6.2. Users can easily add custom pipelines by following the instructions in the project repository.

3.3 Pre-generated Outputs

In addition to interactive generation, TabGenie allows users to visualize static pre-generated outputs. These are loaded in the JSONL[4] format from a specified directory and displayed similarly to model-generated outputs from the interactive mode. Multiple outputs can be displayed alongside a specific example, allowing to compare outputs from multiple systems.

4 Developer Tools

TabGenie also provides a developer-friendly interface: Python bindings (§4.1) and a command-line interface (§4.2). Both of these interfaces aim to simplify dataset preprocessing in downstream tasks. The key benefit of using TabGenie is that it provides streamlined access to data in a consistent format, removing the need for dataset-specific code for extracting information such as table properties, references, or individual cell values.

[4]https://jsonlines.org

4.1 Python Bindings

TabGenie can be integrated in other Python codebases to replace custom preprocessing code. With a *single unified interface* for all the datasets, the TabGenie wrapper class allows to:

- load a dataset from the Huggingface Datasets or from a local folder,
- access individual table cells and their properties,
- linearize tables using pre-defined or custom functions,
- prepare the Huggingface `Dataset` objects for downstream processing.

TabGenie can be installed as a Python package, making the integration simple and intuitive. See §6.1 for an example usage of the TabGenie Python interface.

4.2 Command-line Tools

TabGenie supports several basic commands via command line.

Run The `tabgenie run` command launches the local web server, mimicking the behavior of `flask run`. Example usage:

```
tabgenie run --port=8890 --host="0.0.0.0"
```

Export The `tabgenie export` command enables batch exporting of the dataset. The supported formats are `xlsx`, `html`, `json`, `txt`, and `csv`. Except for `csv`, table properties can be exported along with the table content. Example usage:

```
tabgenie export --dataset "webnlg" \
  --split "dev" \
  --out_dir "export/datasets/webnlg" \
  --export_format "xlsx"
```

Export can also be done in the web interface.

Spreadsheet For error analysis, it is common to select N random examples from the dataset along with the system outputs and manually annotate them with error categories (see §6.3). The `tabgenie sheet` command generates a suitable spreadsheet for this procedure. Example usage:

```
tabgenie sheet --dataset "webnlg" \
  --split "dev" \
  --in_file "out-t5-base.jsonl" \
  --out_file "analysis_webnlg.xlsx" \
  --count 50
```

5 Implementation

TabGenie runs with Python >=3.8 and requires only a few basic packages as dependencies. It can be installed as a stand-alone Python module from PyPI (`pip install tabgenie`) or from the project repository.

Backend The web server is based on `Flask`,[5] a popular lightweight Python-based web framework. The server runs locally and can be configured with a YAML[6] configuration file. On startup, the server loads the data using the `datasets`[7] package. To render web pages, the server uses the `tinyhtml`[8] package and the Jinja[9] templating language. We provide details on the computational and memory requirements in Appendix D.

Frontend The web frontend is built on HTML5, CSS, Bootstrap,[10] JavaScript, and jQuery.[11] We additionally use the D3.js[12] library for visualizing the structure of data in graph-to-text datasets. To keep the project simple, we do not use any other major external libraries.

6 Case Studies

In this section, we outline several recipes for using TabGenie in D2T generation research. The instructions and code samples for these tasks are available in the project repository.

6.1 Table-To-Text Generation

Application Finetuning a sequence-to-sequence language model for table-to-text generation in PyTorch (Paszke et al., 2019) using the Huggingface Transformers (Wolf et al., 2020) framework.

Process In a typical finetuning procedure using these frameworks, the user needs to prepare a `Dataset` object with tokenized input and output sequences. Using TabGenie, preprocessing a specific dataset is simplified to the following:

```
from transformers import AutoTokenizer
import tabgenie as tg

# instantiate a tokenizer
tokenizer = AutoTokenizer.from_pretrained(...)

# load the dataset
tg_dataset = tg.load_dataset(
    dataset_name="totto"
)

# preprocess the dataset
hf_dataset = tg_dataset.get_hf_dataset(
    split="train",
    tokenizer=tokenizer
)
```

The function `get_hf_dataset()` linearizes the tables (the users may optionally provide their custom linearization function) and tokenizes the inputs and references.

For training a single model on multiple datasets in a multi-task learning setting (Xie et al., 2022), the user may preprocess each dataset individually, prepending a dataset-specific task description to each example. The datasets may then be combined using the methods provided by the `datasets` package.

Demonstration For running the baselines, we provide an example script, which can be applied to any TabGenie dataset and pre-trained sequence-to-sequence model from the `transformers` library. For multi-task learning, we provide an example of joint training on several datasets with custom linearization functions. We run the example scripts for several datasets and display the resulting generations in the application demo. Details on the fine-tuned models can be found in Appendix B.

6.2 Interactive Prompting

Application Observing the impact of various inputs on the outputs of a LLM prompted for table-to-text generation.

Process The user customizes the provided `model_api` pipeline to communicate with a LLM through an API. The API can communicate either with an external model (using e.g. OpenAI API[13]), or with a model running locally (using libraries such as FastAPI[14]). The user then interacts with the model through TabGenie web interface, modifying the prompts, highlighted cells, and table content (see §3.2).

[5] https://pypi.org/project/Flask/
[6] https://yaml.org
[7] https://pypi.org/project/datasets/
[8] https://pypi.org/project/tinyhtml/
[9] https://jinja.palletsprojects.com/
[10] https://getbootstrap.com/
[11] https://jquery.com
[12] https://d3js.org
[13] https://openai.com/api/
[14] https://fastapi.tiangolo.com

Demonstration We provide an interactive access to the instruction-tuned Tk-Instruct LLM (Wang et al., 2022) in the project live demo. The user can use the full range of possibilities included in the interactive mode, including customizing the prompt and the input data.[15] The interface is shown in Appendix C.

6.3 Error Analysis

Application Annotating error categories in the outputs from a table-to-text generation model.

Process The user generates the system outputs (see §6.1) and saves the outputs for a particular dataset split in a JSONL format. Through the command-line interface, the user will then generate a XLSX file which can be imported in any suitable office software and distributed to annotators for performing error analysis.

Demonstration We provide instructions for generating the spreadsheet in the project documentation. See Appendix C for a preview of the spreadsheet format.

7 Related Work

7.1 Data Loading and Processing

As noted throughout the work, Huggingface Datasets (Lhoest et al., 2021) is a package commonly used for data loading and preprocessing. TabGenie serves as a wrapper on top of this package, providing additional abstractions and better data visualization for D2T generation datasets.

DataLab (Xiao et al., 2022) is another platform for working with NLP datasets. Similarly to Huggingface Datasets, this platform has much broader focus than our package. Besides data access, it offers fine-grained data analysis and data manipulation tools. However, it has limited capabilities of visualizing the input data or interactive generation and at present, it does not cover the majority of datasets available in TabGenie.

PromptSource (Bach et al., 2022) is a framework for constructing prompts for generative language models using the Jinja templating language. It can be used both for developing new prompts and for using the prompts in downstream applications.

Several tools have been developed for comparing outputs of language generation systems (notably for machine translation) such as CompareMT (Neubig et al., 2019) or Appraise (Federmann, 2018), but the tools do not visualize the structured data.

7.2 Interactive D2T Generation

Platforms for interactive D2T generation have been primarily limited to commercial platforms, such as Arria,[16] Automated Insights,[17] or Tableau Software[18] (formerly Narrative Science). These platforms focus on proprietary solutions for generating business insights and do not provide an interface for research datasets. Dou et al. (2018) present Data2Text Studio, a set of developer tools for building custom D2T generation systems, but their software package currently does not seem to be publicly available.

7.3 Table-To-Text Generation

Although pre-trained sequence-to-sequence models have been found to be effective for D2T generation (Kale and Rastogi, 2020; Xie et al., 2022), they have difficulties with handling the input structure, generation diversity, and logical reasoning. Multiple works have tried to address these issues. For a comprehensive review of the field, we point the interested reader to the recent survey of Sharma et al. (2022).

8 Conclusion

We presented TabGenie, a multifunctional software package for table-to-text generation. TabGenie bridges several gaps including visualizing input data, unified data access, and interactive table-to-text generation. As such, TabGenie provides a comprehensive set of tools poised to accelerate progress in the field of D2T generation.

Limitations

For some D2T generation inputs, the tabular structure may be inappropriate. This involves hierarchical tree-based structures, bag-of-words, or multimodal inputs (Balakrishnan et al., 2019; Lin et al., 2019; Krishna et al., 2017). Due to deployment issues, TabGenie also does not include large synthetic datasets (Agarwal et al., 2021; Jin et al.,

[15] Note that using the model for the task of table-to-text generation is experimental and may not produce optimal outputs. The model should also not be used outside of demonstration purposes due to our limited computational resources.

[16] https://www.arria.com

[17] https://automatedinsights.com

[18] https://www.tableau.com

2020). TabGenie is currently in early development stages, which is why it primarily targets the research community.

Ethical Impact

The table-to-text generation datasets may contain various biases or factually incorrect outputs, which may be further reproduced by the table-to-text generation models. Although our software package is designed to help to examine and eliminate the biases and errors, we cannot guarantee the correctness of the processed outputs.

As TabGenie is an open-source software package with a permissive license, we do not control its downstream applications. We advocate using it for responsible research with the aim of improving natural language generation systems.

Acknowledgements

This research was supported by the European Research Council (Grant agreement No. 101039303 NG-NLG), the Charles University projects GAUK 40222 and SVV 260698, and the Apple NLU Research Grant for Heriot-Watt University and Charles University. Garanina's work is supported by the Erasmus Mundus Master's Programme "Language and Communication Technologies".[19] We acknowledge the utilization of resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth and Sports project No. LM2018101) and Peregrine HPC cluster (University of Groningen).

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-enhanced Language Model Pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3554–3565.

Stephen H. Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M. Saiful Bari, Thibault Févry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged Saeed AlShaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir R. Radev, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 - System Demonstrations, Dublin, Ireland, May 22-27, 2022*, pages 93–104.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained Decoding for Neural NLG From Compositional Representations in Task-oriented Dialogue. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 831–844.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-Text: Describing Table Region With Natural Language. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5020–5027.

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, ACL 2020, Online, July 5-10, 2020*, pages 7929–7942.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020b. Logic2Text: High-fidelity Natural Language Generation From Logical Forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2096–2111.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A Hierarchical Table Dataset for Question Answering and Natural Language Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1094–1110.

Anthony Colas, Ali Sadeghian, Yue Wang, and Daisy Zhe Wang. 2021. EventNarrative: A Large-scale Event-centric Dataset for Knowledge Graph-to-text Generation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.*

Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. 2018. Data2Text Studio: Automated Text Generation From Structured Data. In *Proceedings of the 2018 Conference on Empirical*

[19] `https://lct-master.org`

Methods in Natural Language Processing: System Demonstrations, pages 13–18, Brussels, Belgium.

Ondrej Dusek, David M. Howcroft, and Verena Rieser. 2019. Semantic Noise Matters for Neural Natural Language Generation. In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*, pages 421–426.

Christian Federmann. 2018. Appraise Evaluation Framework for Machine Translation. In *COLING 2018, The 27th International Conference on Computational Linguistics: System Demonstrations, Santa Fe, New Mexico, August 20-26, 2018*, pages 86–88.

Thiago Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 Bilingual, Bi-Directional Webnlg+ Shared Task Overview and Evaluation Results (Webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*.

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, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 124–133.

Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *J. Artif. Intell. Res.*, 61:65–170.

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur P. Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM Benchmark: Natural Language Generation, Its Evaluation and Metrics. *CoRR*, abs/2102.01672.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *CoRR*, abs/2202.06935.

Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. TableGPT: Few-shot Table-to-text Generation With Table Structure Reconstruction and Content Matching. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1978–1988.

Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. 2020. GenWiki: A Dataset of 1.3 Million Content-sharing Text and Graphs for Unsupervised Graph-to-text Generation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2398–2409.

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, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 97–102.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq R. Joty. 2022. Chart-to-Text: A Large-scale Benchmark for Chart Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4005–4023.

Johannes Kehrer and Helwig Hauser. 2013. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *IEEE Trans. Vis. Comput. Graph.*, 19(3):495–513.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.*, 123(1):32–73.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation From Structured Data With Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1203–1213.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*

Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021, pages 175–184.

Bill Yuchen Lin, Ming Shen, Yu Xing, Pei Zhou, and Xiang Ren. 2019. CommonGen: A Constrained Text Generation Dataset Towards Generative Commonsense Reasoning. *CoRR*, abs/1911.03705.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022. PLOG: Table-to-logic Pretraining for Logical Table-to-text Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5531–5546.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Learning to Reason for Text Generation From Scientific Tables. *CoRR*, abs/2104.08296.

Linyong Nan, Dragomir R. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 432–447.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. Compare-Mt: A Tool for Holistic Comparison of Language Generation Systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 35–41.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions With Human Feedback. In *NeurIPS*.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1173–1186.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-shot Task Generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter Open-access Multilingual Language Model. *CoRR*, abs/2211.05100.

Mandar Sharma, Ajay Gogineni, and Naren Ramakrishnan. 2022. Innovations in Neural Data-to-text Generation. *CoRR*, abs/2207.12571.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards Table-to-text Generation With Numerical Reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1451–1465.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. MVP: Multi-task Supervised Pre-training for Natural Language Generation. *CoRR*, abs/2206.12131.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2020. SportSett:basketball - A Robust and Maintainable Data-set for Natural Language Generation. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 32–40, Santiago de Compostela, Spain.

Chris van der Lee, Chris Emmery, Sander Wubben, and Emiel Krahmer. 2020. The CACAPO Dataset: A Multilingual, Multi-domain Dataset for Neural Pipeline and End-to-end Data-to-text Generation. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 68–79.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Stephanie Schoch, Craig Thomson, and Luou Wen. 2023. Barriers and Enabling Factors for Error Analysis in NLG Research. *Northern European Journal of Language Technology*, 9(1). Number: 1.

Emiel van Miltenburg, Miruna-Adriana Clinciu, Ondrej Dusek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of Errors in NLG Output, and What to Do About It. In *Proceedings of the 14th International Conference on Natural Language Generation, INLG 2021*, pages 140–153, Aberdeen, Scotland, UK.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45.

Yang Xiao, Jinlan Fu, Weizhe Yuan, Vijay Viswanathan, Zhoumianze Liu, Yixin Liu, Graham Neubig, and Pengfei Liu. 2022. DataLab: A Platform for Data Analysis and Intervention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 182–195, Dublin, Ireland.

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. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 602–631.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries From Natural Language Using Reinforcement Learning. *CoRR*, abs/1709.00103.

A ToTTo Header Highlighting

The original ToTTo dataset does not provide explicit highlighting for the row and column headers of the corresponding cells. The released table linearization script includes header linking, but the header extraction heuristic is prone to errors in some cases (see, e.g., tables 310 and 838 in the ToTTo development split). Therefore, we implemented a custom algorithm for incorporating header highlighting into our dataset version:

1. process raw ToTTo input: restore the rectangular representation of the table including the merged cells;
2. separate given headers into the column and row headers;
3. for each highlighted cell, highlight all column headers above it and all row headers to the left.

Out of 50 randomly selected tables with highlights different from the available ones, our version was correct and the original version incorrect in 36 cases. Out of the remaining 14 cases, the original version is correct while ours is not in 5 cases, and both versions are incorrect in 9 cases. However, given the complex structure of some tables, our algorithm can sometimes fail to capture relevant row headers or mark several extra cells, which we plan to address in the future.

B Fine-tuned models

For the demo purposes, we have fine-tuned the following models using our example scripts:

- `t5-small` for Chart-To-Text, LogicNLG, ToTTo, WikiTableText;
- `t5-base` for DART, E2E, WebNLG;
- `t5-base` in a prefix-based multi-task setup on E2E and WebNLG, using custom linearization functions.

All models (individual and multi-task) were fine-tuned using `transformers` library. The parameters are the following:

- Epochs: 30 for individual models and 15 for multi-task,
- Patience: 5 epochs,
- Batch size: 16,
- Optimizer: AdamW,
- Learning rate: 1e-4,
- Weight decay: 0,
- AdamW betas: 0.9, 0.999,
- Maximum input length: 512,
- Maximum output length: 512,
- Generation beam size: 3.

C User Interface

Figure 3 shows the interactive mode in the TABGENIE web interface. Figure 4 shows the spreadsheet for manual annotations generated using TABGENIE.

D System Requirements

Computational Requirements TABGENIE can run on a single-threaded CPU, although multiple threads can speed-up the initial dataset preprocessing.

Memory Requirements After pre-loading all the currently featured development sets into memory, TABGENIE consumes around 1 GB RAM.

Disk space Downloading all the datasets requires around 4 GB of disk space. The directory used for caching the datasets can be set using the `HF_DATASETS_CACHE` environment variable.

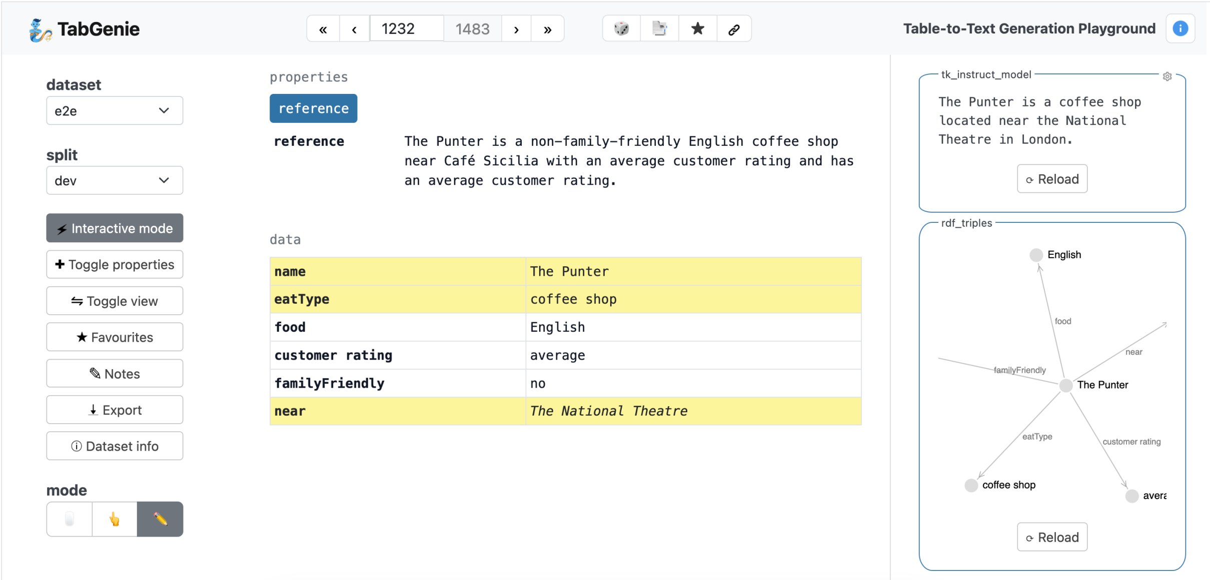

Figure 3: The interactive mode of the web interface in which the user (1) highlighted specific cells (the cells with the yellow background), (2) edited the input in one of the cells (“*Café Sicilia*” → “*the National Theatre*”), (3) re-generated the model output (see the top right panel). The figure also shows the graph visualization of the input key-value pairs.

	A	B	C	D	E	F	G	H
1	table_id	notes	property_name	property_value	table			
2	1309		reference	Adenan Satem was b	subject	predicate	object	
3			prediction	Adenan Satem was b	Abdul Taib Mahmud	successor	Adenan Satem	
4					Adenan Satem	birth place	Japanese occupation of British Borneo	
5					Abdul Taib Mahmud	residence	Sarawak	
6					Abdul Taib Mahmud	party	Barisan Ra'ayat Jati Sarawak	
7								
8	228		reference	Asam pedas is a food	subject	predicate	object	
9			prediction	Asam pedas is a food	Asam pedas	country	Malaysia	
10								
11	51		reference	Aleksandra Kovač's g	subject	predicate	object	
12			prediction	Aleksandra Kovac pe	Aleksandra Kovač	genre	Soul music	
13								
14	1518		reference	Chinabank, a publicly	subject	predicate	object	
15			prediction	Chinabank was found	Chinabank	founding date	1920-08-16	
16					Chinabank	net income	15100000000	
17					Chinabank	number of locations	295	
18					Chinabank	foundation place	Manila	
19					Chinabank	type	Public company	
20								
21	563		reference	The main product of	subject	predicate	object	
22			prediction	Hypermarcas, located	Hypermarcas	product	Drugs	
23					Hypermarcas	location	São Paulo	
24								
25	501		reference	The Asher and Mary	subject	predicate	object	
26			prediction	Asher and Mary Isab	Asher and Mary Isab	location	U.S. Route 83	
27					Asher and Mary Isab	National register of h	88002539	

Figure 4: The spreadsheet for manual annotations with a random sample of system outputs exported using TabGenie.