Transformers for Tabular Data Representation: A Tutorial on Models and Applications

Gilbert Badaro
EURECOM
Biot, France
gilbert.badaro@eurecom.fr

Paolo Papotti
EURECOM
Biot, France
paolo.papotti@eurecom.fr

ABSTRACT

In the last few years, the natural language processing community witnessed advances in neural representations of free texts with transformer-based language models (LMs). Given the importance of knowledge available in relational tables, recent research efforts extend LMs by developing neural representations for tabular data. In this tutorial, we present these proposals with two main goals. First, we introduce to a database audience the potentials and the limitations of current models. Second, we demonstrate the large variety of data applications that benefit from the transformer architecture. The tutorial aims at encouraging database researchers to engage and contribute to this new direction, and at empowering practitioners with a new set of tools for applications involving text and tabular data.

PVLDB Reference Format:
Gilbert Badaro and Paolo Papotti. Transformers for Tabular Data Representation: A Tutorial on Models and Applications. PVLDB, 15(12): 3746 - 3749, 2022.
doi:10.14778/3554821.3554890

1 INTRODUCTION

Several efforts are researching how to represent tabular data with neural models for natural language processing (NLP) and database (DB) applications. These models enable effective solutions that go beyond the limits of traditional declarative specifications built around first order logic and SQL. Examples include answering queries expressed in natural language [16, 19, 31], performing natural language inference such as fact-checking [7, 18, 35], semantic parsing [36, 37], retrieving relevant tables [20, 25, 33], understanding table metadata [8, 11, 29], data integration [6, 22], data to text generation [32] and data imputation [8, 17]. Since these applications involve both structured data and natural language, they are built on new data representations and architectures that go beyond the traditional DB approaches.

Neural Approaches. Transformer-based models, based on the attention mechanism, have been successfully used to develop pre-trained language models (LMs) such as BERT [9], and RoBERTa [24]. These LMs have revolutionized the NLP field with stunning results in the target textual tasks such as sentiment analysis compared to traditional techniques [2, 3]. However, transformers have proven to be able to go beyond text and have been used successfully as well on visual [10] and audio [14] data. Following this trend, transformers have started to gain popularity for developing representations for *tabular data*.

This tutorial focuses on the core problem of rendering the transformer architecture 'data structure aware' and it relates design choices and contributions to a large set of downstream tasks. The attendees can learn about the different ways to use transformers according to the target applications.

Example. When adopting a transformer-based approach, the choices range from adopting existing pre-trained models, created starting from millions of tables, to building solutions from scratch. As an example of an architecture with transformers, consider Fig. 1. Language models are created with the top pipeline (1). In BERT [9], for example, a large corpus of documents is processed with self-supervising tasks to create the model that is then used to build text-centric applications. The creation of the model is expensive, but the final model can be used by any practitioner with an online Python notebook. The most popular way to build an application is to *fine-tune* such model with a small number of specific examples, e.g., classification of documents or sentiment analysis. This is depicted in the bottom pipeline (2).

Moving from text to tabular data, a corpus of tables is used in some approaches to create a pre-trained model which "understands" the tabular format (1). A target application can now use this model to address a downstream task (2). Both in (1) and (2), the table is first serialized and concatenated to its content to feed it as input to the transformers. For example, in (1) the training data can be a large corpus of tables extracted from Wikipedia. (2) is using the pre-trained model to directly answer a query expressed in natural language over a given table. The input of the examples is a table, along with its header "Population in Million by Country" as context, and the question about France population. The desired output is the highlighted cell in the given table. When the pre-trained model does not suffice for the task, it can be fine-tuned with few examples (2). In some cases, the model is pre-trained from scratch (1) to exploit new extensions on the typical transformer architecture to account for the tabular structure, which is different and sometimes richer than the traditional free text.

Outline. Our tutorial consists of three main parts. In the first part, we formalize the problem by providing general definitions and highlight the most common approaches to tackle the neural representation of tabular data (Section 2.1). In the second part, we describe and contrast the most recent works according to five

Proceedings of the VLDB Endowment, Vol. 15, No. 12 ISSN 2150-8097.
doi:10.14778/3554821.3554890

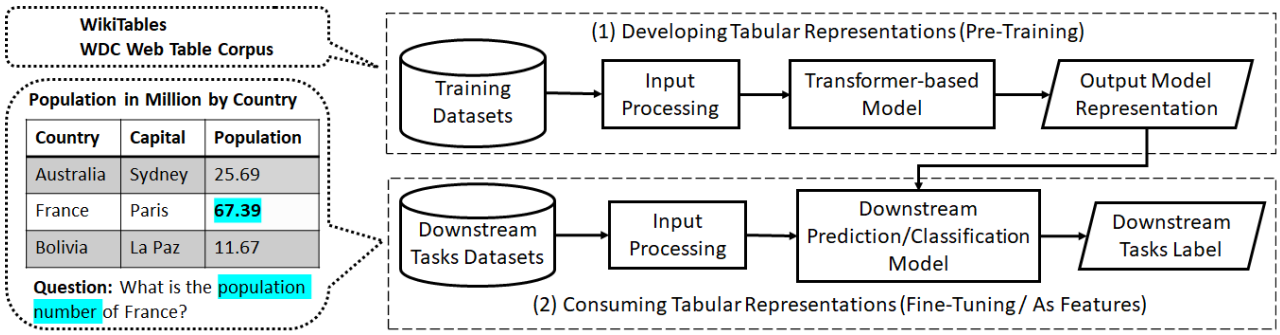

Figure 1: The overall framework for developing and consuming neural representations for tabular data with a data sample. Wikitables or WDC Table Corpus are typically used in (1). In this example, the table along with its header, the additional question and the highlighted answer are used in (2) for a Question Answering downstream task. Both processes combine the serialized table data with natural language text, namely *context*, such as titles, captions, and questions.

dimensions: datasets, data pre-processing, extensions to the transformer architecture, output characteristics, and usage (Sections 2.2 and 2.3). Finally, we discuss limitations of existing works and open research problems tailored for a DB audience (Section 2.4).

2 OUTLINE OF THE TUTORIAL

The tutorial follows the following outline.

2.1 Neural Data Representation

We start by providing an overview of the main use cases exploiting language models with transformers. We also provide a summary on the vanilla transformer-based language model since many of the efforts discussed in Section 2.3 present extensions to that architecture. We then introduce the analogy with tabular data by giving a general problem definition and a high-level overview of a generalized solution. Finally, we show examples of different tasks where the use of those representations proved to achieve state-of-the-art accuracy results for applications involving tabular data and text. For one task, we also demonstrate a live demo with a pre-trained model in an online Python environment[1]. This part covers:

(1) Transformer-based Language Models (LMs): summary and examples of existing models such as BERT [9].
(2) Neural Representation of Tabular Data: Problem Definition and Generalized Solution.
(3) Applications and Target Tasks:
    - Natural Language Inference: fact-checking, text entailment.
    - Question Answering (with Hugging Face TAPAS demo).
    - Semantic Parsing: Text-to-SQL.
    - Table Retrieval.
    - Table Metadata Prediction: detecting column types, relations, header cells; entity resolution and linking, column name prediction.
    - Data Imputation: cell population.

Take-away: attendees become familiar with Transformers architecture and typical existing language models. They also get a feel of the versatility of neural representations for tabular data in multiple data-centric applications.

[1]https://huggingface.co/google/tapas-base-finetuned-wtq

2.2 Characterization of the Methods

In the second part, we detail the dimensions to describe and categorize the different proposals. We focus our tutorial on the extensions to the original transformer architecture for developing representations of relational tables. While several solutions have contributed to the transformer original architecture to better represent tabular data, the alternative innovations to model and consume the encoded data are scattered over the process. We aim at bringing clarity in this space by providing an overview with a set of dimensions that let us highlight the main ideas and trends spanning the different proposals. We use five dimensions summarized below. More details on the proposed dimensions can be found in our survey paper [4].

(1) Training Datasets: comparative summary of characteristics of datasets used for learning the table data representations along with some representative samples. Four datasets are typically exclusively used for pre-training, e.g., WikiTables [5], WDC Web Table Corpus [21]. The majority of the datasets include extra manual annotations to enable their usage for fine-tuning or evaluation. Examples of such datasets include TabFact [7], WikiSQL [39], FEVEROUS [1] and SPIDER [38].
(2) Input Processing: textual and tabular pre-processing steps of the training data prior to feeding it to the neural network.
    - Data Retrieval and Filtering: to meet the limits of transformer based architectures or to reduce noisy representations.
    - Table Serialization: linearizing the table to feed it as input to the neural network.
    - Context and Table Concatenation: the context can consist of table metadata, table descriptions, captions, and questions whose answer can be found in the corresponding table. The type and amount of context depend on the target application.
(3) Model Architecture and Training: different model customizations are performed on typical LMs to accommodate tabular data. These can be grouped as changes or extensions on the input/output layers or on the internals of the model: Rows and Columns specific Encodings, Table Structure Aware Representation, Selection of Base LM Model, Direction of

Attention, Pre-training Objectives, Addition of CLS Layers, and Fine-tuning Objectives.
(4) Output Model Representation: different granularity of representations of table content.
(5) Fine-tuning Representations for Downstream Tasks.

Take-away: Following this second part, the audience can grasp the characteristics of the different existing solutions and classify upcoming ones along the same dimensions for easier comparison.

2.3 Latest Works in the Field

After detailing the dimensions in Section 2.2, we analyze the latest research efforts in the field based on those dimensions. We briefly discuss how 20 surveyed works [7, 8, 11–13, 15–17, 20, 23, 25, 29–31, 33–37] address the five dimensions following the framework in Fig. 1.

Most works opt for pre-training ((1) in Fig. 1) followed by fine-tuning and consuming the representations to tackle downstream tasks ((2) in Fig. 1). A few exceptions either fine-tune existing LMs or use them as part of their features set [11, 20, 29]. For developing tabular representations, most of the works aim at supporting significantly large datasets, up to millions of tuples, by combining multiple datasets for more accurate generalized representations. The steps in the *Input Processing* part (first module for both (1) and (2) in Fig. 1) are typically set without exploring and comparing the different possible variations except for a few cases where authors evaluate different settings such as row vs. column serialization and context followed by serialized table vs. table appended by context [7, 32].

The component that makes the major difference among the surveyed works is *Transformer-based Model* through the customization and extensions on the vanilla transformer (second module in (1) in Fig. 1). The main objective of the customization is to preserve the 2-dimensional tabular data characteristics while linearizing it into 1-dimensional space as the free text one. While these extensions can be grouped based on the level they are applied on, i.e. input, internal and output levels, their application details remain more or less unique. For instance, at the input level, to account for the position of the cells, Herzig et al. add extra dimensions to the embedding vector to account for cell, row, and column positions [16], while Wang et al. uses a bi-dimensional coordinate tree [34]. At the internal level, modifications concern the attention mechanism to further emphasize the tabular structure. For example, Yin et al. use vertical self-attention layers [36] while Eisenschlos et al. employ sparse attention to efficiently attend to rows and columns [12]. At the output level, the extensions are tailored for the intended downstream tasks and they are manifested mostly by the addition of classification layers.

The *Output Model Representation* (third module in (1) in Fig. 1) has different granularity depending on the intended downstream task, i.e., cell, row, column or table representations. For instance, Herzig et al. generate cell representations for the QA task, Wang et al. use table representations to facilitate table retrieval (TR) task, and Liu et al. utilize token embeddings for semantic parsing. These representations are then either fine-tuned using labeled downstream tasks datasets [25] or utilized as features of training data points [11].

Take-away: After attending the third part of the tutorial, the audience can match a target application to the most effective solution. They also have a good understanding of the main technical challenges from a data perspective.

2.4 Open Challenges & Conclusion

While there has been progress in developing and consuming tabular data representations, several challenges remain unaddressed. We discuss these directions with the audience to show where the DB community can have the greatest impact for this problem. Similar to other efforts, the challenges of interpretability, the need of more significant error analysis, and model efficiency are also applicable for the case of developing and consuming neural representations for relational data. Some systems expose a justification of their model output [12, 16, 25, 31, 35], but the majority does not, and model usage remains a black box. More specifically to relational data, complex queries remain difficult to handle especially when they involve joining tables. Last but not least, in contrast to what has been done for LMs for text [26], there is a lack in terms of benchmarking data representations. A new family of data-driven basic tests should be designed to measure the consistency of the data representation.

3 TUTORIAL: TYPE, AUDIENCE, DIVERSITY, ETHICS AND PREREQUISITES

This tutorial covers the latest developments in the neural representation for relational data and their application. Unlike the tutorial [19] of Katsogiannis-Meimarakis and Koutrika that focuses specifically on solutions that address the semantic parsing task, i.e., converting text to query, we cover a wider scope of data-centric tasks addressed thanks to the versatility of transformers and language models. It is of interest to researchers looking to integrate knowledge from structured data, namely tables, in addition to unstructured data into the different mentioned downstream tasks. The tutorial is not only for practitioners working on applications with the English language, thanks to the the multilingual LMs that are utilized as basis to develop the representations for relational data. For more details about the characterization of the transformer-based models for neural representation of database tables, we refer the readers to our survey paper [4].

The use of large-scale Transformers requires a lot of computations and GPUs/TPUs for training, which contributes to global warming [27, 28]. We stress this orthogonal issue and possible approaches to mitigate it in the tutorial. The datasets used do not include private data.

Prior knowledge of machine learning is not mandatory as we deliver an introductory overview of the transformer architecture in the first part (Section 2.1).

4 PRESENTERS

Gilbert Badaro is a Post-Doc fellow at the Data Science Department, EURECOM (France) since April 2021. He received his Ph.D. in Electrical & Computer Engineering from the American University of Beirut (Lebanon) in 2020. His research expertise is in NLP and ML. He has authored more than 20 publications and is a reviewer in journals and conferences, such as ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP). At EURECOM, he works on fact-checking of statistical claims and

on using neural representation for database tables to develop text verification systems.

Paolo Papotti is an Associate Professor at EURECOM (France) since 2017. He got his PhD from Roma Tre University (Italy) in 2007 and had research positions at the Qatar Computing Research Institute (Qatar) and Arizona State University (USA). His research is focused on data management and information quality, with recent contributions in computational fact-checking and pre-trained language models. He has authored more than 100 publications and his work has been recognized with two "Best of the Conference" citations (SIGMOD 2009, VLDB 2016), two best demo award (SIGMOD 2015, DBA 2020), and two Google Faculty Research Award (2016, 2020). He is the associate editor for PVLDB and ACM JDIQ.

REFERENCES

[1] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. In *NeurIPS Datasets and Benchmarks Track (Round 1)*.

[2] Gilbert Badaro, Ramy Baly, Hazem Hajj, Wassim El-Hajj, Khaled Bashir Shaban, Nizar Habash, Ahmad Al-Sallab, and Ali Hamdi. 2019. A survey of opinion mining in Arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18, 3 (2019), 1–52.

[3] Gilbert Badaro, Hazem Hajj, and Nizar Habash. 2020. A link prediction approach for accurately mapping a large-scale Arabic lexical resource to English WordNet. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 19, 6 (2020), 1–38.

[4] Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. 2021. Transformers for Tabular Data Representation: A survey of models and applications. EURECOM Technical Report, October 2021: https://www.eurecom.fr/publication/6721.

[5] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: Entity linking in web tables. In *International Semantic Web Conference*. Springer, 425–441.

[6] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating embeddings of heterogeneous relational datasets for data integration tasks. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1335–1349.

[7] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkeJRhNYDH

[8] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table understanding through representation learning. *Proceedings of the VLDB Endowment* 14, 3 (2020), 307–319.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NACL: HLT*. ACL, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[11] Lun Du, Fei Gao, Xu Chen, Ran Jia, Junshan Wang, Jiang Zhang, Shi Han, and Dongmei Zhang. 2021. TabularNet: A Neural Network Architecture for Understanding Semantic Structures of Tabular Data. In *ACM SIGKDD*. 322–331.

[12] Julian Martin Eisenschlos, Maharshi Gor, Thomas Müller, and William W Cohen. 2021. MATE: Multi-view Attention for Table Transformer Efficiency. *arXiv preprint arXiv:2109.04312* (2021).

[13] Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avirup Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing Row and Column Semantics in Transformer Based Question Answering over Tables. In *NACL: HLT*. 1212–1224.

[14] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio Spectrogram Transformer. *arXiv preprint arXiv:2104.01778* (2021).

[15] Jonathan Herzig, Thomas Mueller, Syrine Krichene, and Julian Eisenschlos. 2021. Open Domain Question Answering over Tables via Dense Retrieval. In *NACL: HLT*. 512–519.

[16] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4320–4333.

[17] Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained Representations of Tabular Data. In *NACL: HLT*. 3446–3456.

[18] Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. 2020. Scrutinizer: A Mixed-Initiative Approach to Large-Scale, Data-Driven Claim Verification. *Proc. VLDB Endow.* 13, 11 (2020), 2508–2521.

[19] George Katsogiannis-Meimarakis and Georgia Koutrika. 2021. A Deep Dive into Deep Learning Approaches for Text-to-SQL Systems. In *Proceedings of the 2021 International Conference on Management of Data*. 2846–2851.

[20] Bogdan Kostić, Julian Risch, and Timo Möller. 2021. Multi-modal Retrieval of Tables and Texts Using Tri-encoder Models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*. ACL, Punta Cana, Dominican Republic, 82–91. https://aclanthology.org/2021.mrqa-1.8

[21] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *WWW Companion*. 75–76.

[22] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endow.* 14, 1 (2020), 50–60. https://doi.org/10.14778/3421424.3421431

[23] Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-guang Lou. 2021. TAPEX: Table pre-training via learning a neural SQL executor. *arXiv preprint arXiv:2107.07653* (2021).

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[25] Feifei Pan, Mustafa Canim, Michael Glass, Alfio Gliozzo, and Peter Fox. 2021. CLTR: An End-to-End, Transformer-Based System for Cell-Level Table Retrieval and Table Question Answering. In *ACL System Demonstrations*. 202–209.

[26] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *ACL*. ACL, Online, 4902–4912. https://doi.org/10.18653/v1/2020.acl-main.442

[27] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM* 63, 12 (2020), 54–63.

[28] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13693–13696.

[29] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2021. Annotating Columns with Pre-trained Language Models. *arXiv preprint arXiv:2104.01785* (2021).

[30] Nan Tang, Ju Fan, Fangyi Li, Jianhong Tu, Xiaoyong Du, Guoliang Li, Samuel Madden, and Mourad Ouzzani. 2021. RPT: Relational Pre-trained Transformer Is Almost All You Need towards Democratizing Data Preparation. *Proc. VLDB Endow.* 14, 8 (2021), 1254–1261.

[31] James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. 2021. Database reasoning over text. In *ACL*. 3091–3104.

[32] Enzo Veltri, Donatello Santoro, Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. 2022. Pythia: Unsupervised Generation of Ambiguous Textual Claims from Relational Data. In *SIGMOD - Demo track*. ACM.

[33] Fei Wang, Kexuan Sun, Muhao Chen, Jay Pujara, and Pedro Szekely. 2021. Retrieving Complex Tables with Multi-Granular Graph Representation Learning. In *SIGIR*. ACM, 1472–1482.

[34] Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. TUTA: Tree-based Transformers for Generally Structured Table Pre-training. In *ACM SIGKDD*. 1780–1790.

[35] Xiaoyu Yang and Xiaodan Zhu. 2021. Exploring Decomposition for Table-based Fact Verification. In *EMNLP 2021*. ACL, Punta Cana, Dominican Republic, 1045–1052. https://aclanthology.org/2021.findings-emnlp.90

[36] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. In *ACL*. ACL, Online, 8413–8426. https://doi.org/10.18653/v1/2020.acl-main.745

[37] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, bailin wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, richard socher, and Caiming Xiong. 2021. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. In *International Conference on Learning Representations*. https://openreview.net/forum?id=kyaIeYj4zZ

[38] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *EMNLP*. ACL, Brussels, Belgium, 3911–3921. https://doi.org/10.18653/v1/D18-1425

[39] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103* (2017).