



# Models and Practice of Neural Table Representations

Madelon Hulsebos  
University of Amsterdam  
m.hulsebos@uva.nl

Huan Sun  
The Ohio State University  
sun.397@osu.edu

Xiang Deng  
The Ohio State University  
deng.595@osu.edu

Paolo Papotti  
EURECOM  
papotti@eurecom.fr

## ABSTRACT

In the last few years, the natural language processing community witnessed advances in neural representations of free-form text 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<sup>1</sup>, we present these proposals with three main goals. First, we aim at introducing the potentials and limitations of current models to a database audience. Second, we want the attendees to see the benefit of such line of work in a large variety of data applications. Third, we would like to empower the audience with a new set of tools and to inspire them to tackle some of the important directions for neural table representations, including model and system design, evaluation, application and deployment. To achieve these goals, the tutorial is organized in two parts. The first part covers the background for neural table representations, including a survey of the most important systems. The second part is designed as a hands-on session, where attendees will use their laptop to explore this new framework and test neural models involving text and tabular data.

## CCS CONCEPTS

• **Information systems** → *Relational database model*; • **Computing methodologies** → *Neural networks*; **Natural language processing**.

## KEYWORDS

tables, representation learning, data management

### ACM Reference Format:

Madelon Hulsebos, Xiang Deng, Huan Sun, and Paolo Papotti. 2023. Models and Practice of Neural Table Representations. In *Companion of the 2023 International Conference on Management of Data (SIGMOD-Companion '23)*, June 18–23, 2023, Seattle, WA, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3555041.3589411>

<sup>1</sup>Up to date information and material for the tutorial can be found at: <https://github.com/madelonhulsebos/neural-table-representations-tutorial-2023>



This work is licensed under a Creative Commons Attribution International 4.0 License.

*SIGMOD-Companion '23*, June 18–23, 2023, Seattle, WA, USA  
© 2023 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-9507-6/23/06.  
<https://doi.org/10.1145/3555041.3589411>

## 1 INTRODUCTION

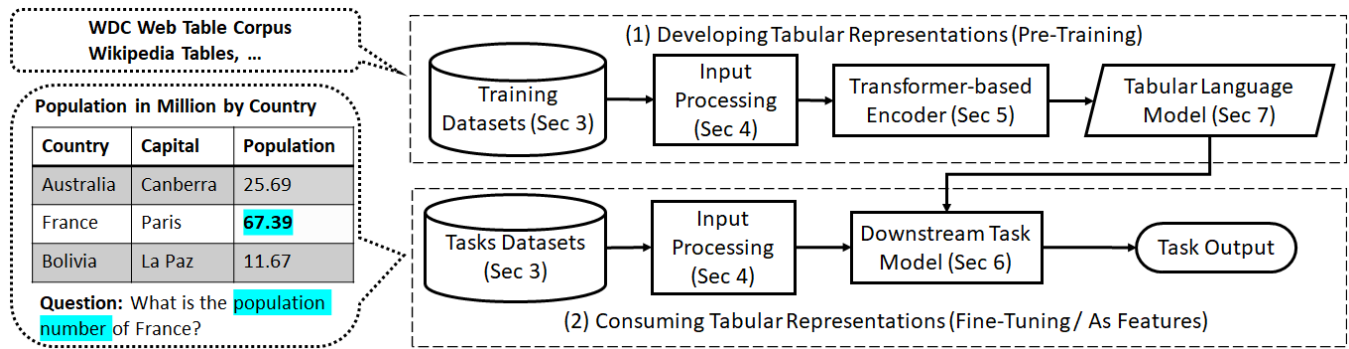
Several efforts are researching how to represent tabular data with neural models for natural language processing (NLP) and database (DB) applications. These models enable effective solutions that go beyond the limits of traditional declarative specifications built around first order logic and SQL. Examples include answering queries expressed in natural language [19, 23, 36], performing natural language inference such as fact-checking [9, 22, 40], semantic parsing [10, 41, 42], retrieving relevant tables [24, 29, 38], understanding table metadata [11, 14, 34], data integration [8, 26], data to text generation [37] and data imputation [11, 21]. 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 [12] and RoBERTa [28]. These LMs have revolutionized the NLP field with stunning results in the target textual tasks, compared to traditional techniques. Transformers have also proven to be able to go beyond text and have been used successfully as well on visual [13] and audio [17] 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 pretrained models, created starting from millions of tables, to building solutions from scratch. As an example of an architecture with transformers, consider Fig. 1 from [3]. Language models are created with the top pipeline (1). In BERT [12], for example, a large corpus of documents is processed with self-supervised 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 pretrained 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



**Figure 1: The overall framework for developing and consuming neural representations for tabular data with a data sample [3]. 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.**

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 pretrained 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 pretrained model does not suffice for the task, it can be fine-tuned with few examples (2). In some cases, the model is pretrained 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.** The tutorial consists of two main parts. The first part is organized as a *survey*, where we first formalize the problem by providing general definitions and highlight the most common approaches to tackle the neural representation of tabular data (Section 2.1). We describe and contrast the most recent works according to five 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).

In the second part, we conduct a *hands-on* session where the audience will be guided in the execution of Python notebooks to explore both vanilla language models and those specialized for tabular data (Section 3). We start with investigating the general data pipeline from input data formatting to processed data (Sections 3.1 and 3.2). We proceed with the actual pretraining procedure (Section 3.3), which is followed by an exercise on fine-tuning and evaluating pretrained models for the downstream task of data imputation (Section 3.4). We close the hands-on session with analyses of the fine-tuned model on table-level, while recapitulating on open issues as discussed in the first part of the tutorial.

## 2 OUTLINE OF THE SURVEY PART

In the first part of the tutorial, we provide an overview of the motivation, characterization of methods, state-of-the-art applications, and open challenges.

### 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 pretrained model in an online Python environment<sup>2</sup>. This part covers:

- (1) Transformer-based Language Models (LMs): summary and examples of existing models such as BERT [12].
- (2) Neural Representation of Tabular Data: Problem Definition and Generalized Solution.
- (3) Applications and Target Tasks:
  - Tabular Natural Language Inference: text entailment, including fact-checking.
  - 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.*

<sup>2</sup><https://huggingface.co/google/tapas-base-finetuned-wtq>

## 2.2 Characterization of the Methods

We then 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 [3].

- (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 pretraining, e.g., WikiTables [6], WDC Web Table Corpus [25]. The majority of the datasets include extra manual annotations to enable their usage for fine-tuning or evaluation. Examples of such datasets include TabFact [9], WikiSQL [44], FEVEROUS [1] and SPIDER [43].
- (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: 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 a sample of the latest research efforts in the field based on those dimensions. We briefly discuss how 20 surveyed works [9, 11, 14–16, 18, 19, 21, 24, 27, 29, 34–36, 38–42] address the five dimensions following the framework in Fig. 1.

Most works opt for pretraining ((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 [14, 24, 34]. 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 [9, 37].

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 [19], while Wang et al. uses a bi-dimensional coordinate tree [39]. 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 [41] while Eisenschlos et al. employ sparse attention to efficiently attend to rows and columns [15]. 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 [29] or utilized as features of training data points [14].

*Take-away: 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 [15, 19, 29, 36, 40], but the majority does not, and model usage remains a black box. A clear, recent example of this challenge is ChatGPT<sup>3</sup>, and all generative models [7], which have strong abilities to answer questions, including queries, but with no clear guarantees of what is factual and what is invented (hallucinated) by the model [4]. More

<sup>3</sup><https://chat.openai.com>

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 [31], 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 OUTLINE OF THE HANDS-ON SESSION

In the second part, we focus on providing hands-on experience with transformer models. Attendees will better understand transformer models for tabular data on a conceptual level through practical exercises, while also becoming familiar with their implementation details and impact in downstream applications. We will demonstrate the general pipelines from input data and data processing to training of various models and fine-tuning for downstream tasks. The structure and content of the hands-on session align with the structure of the first part of the tutorial (Section 2). The hands-on session will be facilitated by notebooks in Google Colaboratory<sup>4</sup>. Throughout the session, we will investigate the synergies and differences between a vanilla language model (BERT [12]), and two specialized tabular models (TURL [11], TAPEX [27] and TaBERT [41]). We will also present small code extracts to connect conceptual understanding of the discussed models to the details on implementation level.

#### 3.1 Off-the-shelf Model Inputs and Outputs

We start with a few example input tables and will use the pretrained models off-the-shelf to obtain numeric vector representations of them. First, we load a given table from a CSV file and format the table such that the respective pretrained model can process it. For BERT, we programmatically linearize the raw table header and values into sequences compatible with BERT, to illustrate basic design choices behind linearization. We proceed with loading the model and using it to encode the formatted table, such that we obtain a vector representation of the table. After each step, we will inspect the intermediate object that is obtained. The code snippet in Figure 2a illustrates this exercise on a high level. Finally, we compare the input formats and output encodings across the three different models (BERT, TAPEX and TaBERT).

*Take-away: This exercise will give a high-level understanding of the input and outputs of these off-the-shelf models, and the differences among them.*

#### 3.2 Table Processing and Encoding

Then, we dive deeper into the inner working of the models, by investigating the formatting of tables for pretraining the transformer models. These table representations are tightly related to the architectural design of these Transformers. For example, the processed table representation for TURL illustrates its linearization procedure as shown in Figure 2b, reflects its vertical attention mechanism, and solicits understanding of the considered pretraining tasks. We will include a few different input processing techniques to illustrate how existing transformer models accommodate structured tabular data as input.

*Take-away: This exercise will give a detailed understanding of how tables are formatted and preprocessed, and how the resulting*

*table encodings relate to architectural designs (e.g. different attention mechanisms).*

#### 3.3 Pretraining and Output Encoding

We will use the two pretraining objectives in TURL [11], namely masked language modeling and masked entity recovery, to demonstrate how to pretrain the transformer model using unlabeled tables and learn to capture the semantic and relational knowledge embedded in the data. We will cover the masking procedure, which involves randomly masking out elements of the tables, and the preparation of target outputs, which the model will be trained to predict. We will proceed with some implementation details of the pretraining, including the hyperparameters, data size, and computing resource requirements. We will also provide utility code to visualize the attention weights and output table encodings, which can help participants understand how the model processes and represents the data.

*Take-away: This exercise will give a detailed understanding of how transformer models can be pretrained over unlabeled table corpus and later used for various downstream applications. The audience will also understand better the working mechanism of the models through visualization.*

#### 3.4 Fine-tuning and Analysis

Finally, we move on to fine-tuning TURL for the downstream task of data imputation for which we use both entity-focused tables from WikiTables [5] and tables from CSV files as in GitTables [20]. We will evaluate the aggregated performance of the fine-tuned model with standard metrics such as F1 on a hold-out set of tables.

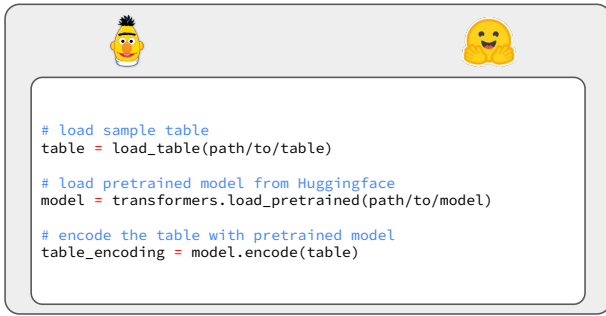
Guided by a few selected test tables and any tables that attendees bring themselves, we will zoom in on cases where the fine-tuned model provides accurate predictions and on cases where it fails. We will also explore a few general challenges of LM-based transformers for tabular data such as accurately representing numeric tables and tables without descriptive headers. These failure cases, provide a bridge to the closing recapitulation on open challenges as discussed in the first part of the tutorial (see Section 2.4).

*Take-away: This exercise will enable attendees to use pretrained models in downstream applications and demonstrate the relatively simple process for doing so. They will also understand the limitations of these models.*

### 4 TYPE, PREREQUISITES AND CONTEXT

**Type.** The tutorial combines a *survey* of the field with a *hands-on* session, and takes 3 hours in total. In the first part of 1.5 hours, we provide an overview of the field and a deep-dive on more recent developments, as explained in Section 2. After a short break, in the second 1.5 hours, we guide participants through a practical session to obtain hands-on experience with Transformer models for tabular data. If needed, the tutorial can be delivered in 1.5 hours by focusing only on the survey or on the hands-on part, according to the recommendation from the chairs. We remark that the first part is self contained, so that attendees not interested in the hands-on can leave at its end. Similarly, it will be possible to join only the second part for attendees who are already familiar with the basic of the transformer and the related models.

<sup>4</sup><https://colab.research.google.com/>



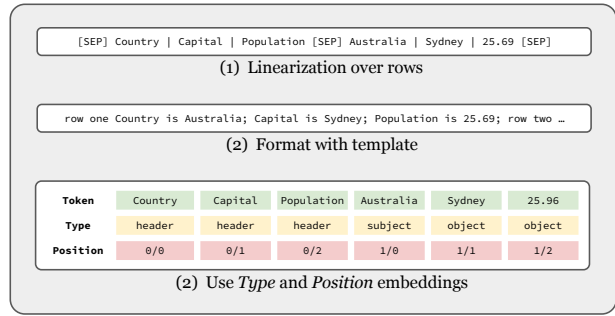
```

# load sample table
table = load_table(path/to/table)

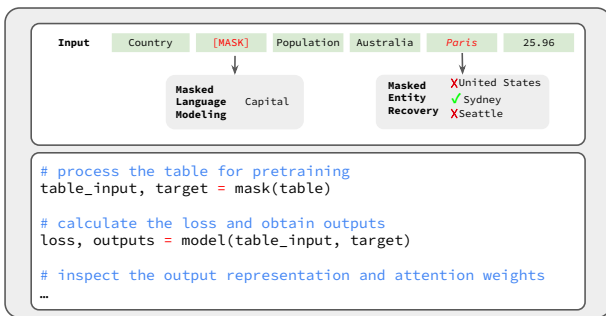
# load pretrained model from Huggingface
model = transformers.load_pretrained(path/to/model)

# encode the table with pretrained model
table_encoding = model.encode(table)
    
```

(a) Off-the-shelf model inputs and outputs. We demonstrate how to use pretrained models off-the-shelf, including BERT, TAPAS and TaBERT.



(b) Table processing and encoding. We demonstrate different techniques used to convert tabular data as input to the transformer model.



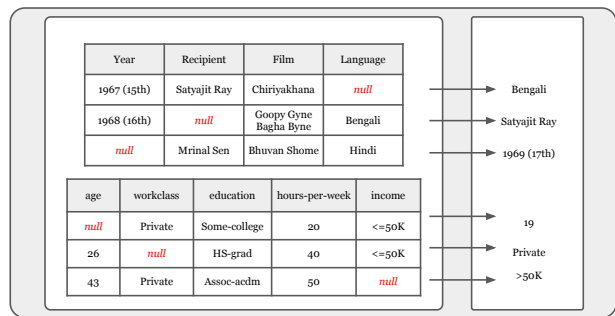
```

# process the table for pretraining
table_input, target = mask(table)

# calculate the loss and obtain outputs
loss, outputs = model(table_input, target)

# inspect the output representation and attention weights
...
    
```

(c) Pretraining and output encoding. We demonstrate the pretraining process using TURL as an example, and dig deeper into the internal works of the transformer model.



(d) Fine-tuning and analysis. We demonstrate the fine-tuning process using data imputation as an example, and conduct some case studies.

Figure 2: Overview of the hands-on session.

**Survey Prerequisites.** The target audience are researchers and practitioners in data management with an interest in the intersection of machine learning, structured data, and applications of neural representations such as data discovery and analysis. Prior knowledge of machine learning is not mandatory as we present an introductory overview of the transformer architecture in the first part (Section 2.1). This introductory overview will be sufficient to follow the rest of the tutorial and we will address technical discussions about the internals of transformers in the Q&A part.

**Hands-on Prerequisites.** During the hands-on part, participants will have access to a configured (Python) development environment that is accessible through the browser in Google Colaboratory<sup>5</sup>. This environment will have required packages installed, and provide a general structure for the tutorial with template code, visual examples, etc. to minimize time to get started. Access to the pre-configured Google Colab notebook requires a Google account. This material can be found at: <https://github.com/madelonhulsebos/neural-table-representations-tutorial-2023>.

**Ethics.** The use of large-scale Transformers requires a lot of computations and GPUs/TPUs for training, which contributes to global

warming [32, 33]. We stress this orthogonal issue and possible approaches to mitigate it in the tutorial. The datasets used do not include private data.

**Difference with Previous Tutorials.** Our tutorial partly overlaps with two of our tutorials that have been offered recently:

- 1) “Transformers for Tabular Data Representation: A Tutorial on Models and Applications” [2] in VLDB 2022 covers only the survey part in our proposal. Moreover, our SIGMOD tutorial aims at covering in more detail a smaller subset of applications. For these applications, we will give more examples and details on how to use them, to enable the attendees to execute them in the hands-on part.
- 2) “From Tables to Knowledge: Recent Advances in Table Understanding” [30] in KDD 2021 is much wider in scope as it covers semantic models and information extraction, such as NER, and did not include hands-on exercises. In this SIGMOD tutorial, we will focus more on the transformer architecture, covering pretraining and fine-tuning, with half of the time dedicated to a hands-on session where attendees can experiment with the models on their laptop.

<sup>5</sup><https://colab.research.google.com/>

## 5 PRESENTERS

**Madelon Hulsebos** is a PhD candidate at the Intelligent Data Engineering Lab of the University of Amsterdam. Previously, she did research at Sigma Computing and the MIT Media Lab after obtaining her BS and MS from Delft University of Technology. Madelon is interested in Table Representation Learning and developing intelligent relational data systems, for which she contributed models, systems, and datasets.

**Xiang Deng** is a Ph.D. candidate in the Department of Computer Science and Engineering at The Ohio State University (OSU). Prior to that, he obtained his B.S. from University of Science and Technology of China. His research interests lie in NLP and data mining, with emphasis on knowledge discovery and utilization from heterogeneous sources. The aim is to build AI-powered data systems that can assist with information acquisition and decision making for regular users as well as domain experts in the digital era. His research has been recognized with ACM SIGMOD Research Highlights (2022) and the Presidential Fellowship (2022-2023) from OSU.

**Huan Sun** is a tenured associate professor in the Department of Computer Science and Engineering at OSU. Before joining OSU, she was a visiting scientist at University of Washington in 2016 and received a Ph.D. from UC Santa Barbara in 2015. Her research interests lie in natural language processing, data mining, and artificial intelligence, with a focus on question answering, semantic parsing, conversational and interactive systems. Her research received the ACM SIGMOD Research Highlight Award and the Best Paper Award from the IEEE International Conference on Bioinformatics and Biomedicine (BIBM). She is a recipient of NSF CAREER Award, Google Research Scholar and Google Faculty Award, OSU Lumley Research Award, and SIGKDD Ph.D. Dissertation Runner-Up Award, among others. Her team TacoBot won third place in the first Alexa Prize TaskBot challenge in 2022.

**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 pretrained 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), three best demo award (SIGMOD 2022, SIGMOD 2015, DBA 2020), and two Google Faculty Research Award (2016, 2020).

## 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 and Paolo Papotti. 2022. Transformers for Tabular Data Representation: A Tutorial on Models and Applications. *Proc. VLDB Endow.* 15, 12 (aug 2022), 3746–3749. <https://doi.org/10.14778/3554821.3554890>
- [3] Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. 2023. Transformers for Tabular Data Representation: A survey of models and applications. *Transactions of the ACL (TACL)* (2023). <https://www.eurecom.fr/~papotti/files/TACL23.pdf>.
- [4] Emily M. Bender, Timnit Gebu, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 610–623. <https://doi.org/10.1145/3442188.3445922>
- [5] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2013. Methods for exploring and mining tables on wikipedia. In *Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics*. 18–26.
- [6] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: Entity linking in web tables. In *International Semantic Web Conference*. Springer, 425–441.
- [7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 <https://arxiv.org/abs/2005.14165>
- [8] 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.
- [9] Wenhui Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyu 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>
- [10] Xiang Deng, Ahmed Hassan, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-Grounded Pretraining for Text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1337–1350.
- [11] 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.
- [12] 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>
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua 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*.
- [14] 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.
- [15] 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).
- [16] Michael Glass, Mustafa Canim, Alfio GlioZZo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avirup Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglaia. 2021. Capturing Row and Column Semantics in Transformer Based Question Answering over Tables. In *NACL: HLT*. 1212–1224.
- [17] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio Spectrogram Transformer. *arXiv preprint arXiv:2104.01778* (2021).
- [18] Jonathan Herzig, Thomas Mueller, Syrine Krichene, and Julian Eisenschlos. 2021. Open Domain Question Answering over Tables via Dense Retrieval. In *NACL: HLT*. 512–519.
- [19] 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.
- [20] Madelon Hulsebos, Çağatay Demiralp, and Paul Groth. 2021. GitTables: A large-scale corpus of relational tables. *arXiv preprint arXiv:2106.07258* (2021).
- [21] Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained Representations of Tabular Data. In *NACL: HLT*. 3446–3456.
- [22] 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.
- [23] 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.
- [24] 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.mrq-a-1.8>
- [25] 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.

- [26] 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>
- [27] 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).
- [28] 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](https://arxiv.org/abs/1907.11692) <http://arxiv.org/abs/1907.11692>
- [29] 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.
- [30] Jay Pujara, Pedro Szekely, Huan Sun, and Muhao Chen. 2021. From Tables to Knowledge: Recent Advances in Table Understanding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) (*KDD '21*). Association for Computing Machinery, New York, NY, USA, 4060–4061. <https://doi.org/10.1145/3447548.3470809>
- [31] 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>
- [32] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM* 63, 12 (2020), 54–63.
- [33] 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.
- [34] 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).
- [35] 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.
- [36] James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. 2021. Database reasoning over text. In *ACL*. 3091–3104.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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>
- [41] 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>
- [42] 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=kyaleYj4zZ>
- [43] 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>
- [44] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103* (2017).