Towards Foundation Models for Relational Databases [Vision Paper]

Liane Vogel
Technical University of Darmstadt

Benjamin Hilprecht
Technical University of Darmstadt

Carsten Binnig
Technical University of Darmstadt & DFKI

Abstract

Tabular representation learning has recently gained a lot of attention. However, existing approaches only learn a representation from a single table, and thus ignore the potential to learn from the full structure of relational databases, including neighboring tables that can contain important information for a contextualized representation. Moreover, current models are significantly limited in scale, which prevents that they learn from large databases. In this paper, we thus introduce our vision of relational representation learning, that can not only learn from the full relational structure, but also can scale to larger database sizes that are commonly found in real-world. Moreover, we also discuss opportunities and challenges we see along the way to enable this vision and present initial very promising results. Overall, we argue that this direction can lead to *foundation models for relational databases* that are today only available for text and images.

1 Introduction

Motivation. Tabular representation learning has recently gained a lot of attention. For example, approaches like RPT (Tang et al., 2021) or TURL (Deng et al., 2020) have shown that they can provide deep contextualized representations by self-supervised pre-training that enable many data engineering tasks such as data cleaning, entity resolution, column type annotation etc. to be efficiently derived from the representation with low overhead.

However, these existing approaches only learn a representation from a single table and thus ignore the potential to learn a representation from the full structure of relational databases, i.e., by taking neighboring tables as well as relationships between tables into account. We argue that being able to process signals from the full relational structure such as neighboring tables enables us to utilize valuable information in relational databases. For example if the information that a neighboring table is called *actors* is available, it is easier for a model to infer that the missing table name of the table under consideration is more likely to be *movies* than *music albums*.

Moreover, many of the recent approaches like TURL or RPT significantly build on language models (LMs) that have been pre-trained on large textual corpora. It was even shown that LMs not adapted for specific data engineering tasks such as entity resolution can already achieve state-of-the-art performance Narayan et al. (2022). Unfortunately, while these approaches based on pre-trained LMs have shown remarkable results and outperform other approaches such as EmbDI or Termite (Cappuzzo et al., 2020; Fernandez and Madden, 2019), they do not scale to large data sizes, since they are often limited by the restrictive input sizes of LMs. While this limitation is negligible for the scenarios that these approaches have been originally developed for (e.g., for web tables or tabular data in CSV files), real-world relational databases, as they can be found in enterprises today, are typically much larger in size (Krüger et al., 2011).

Furthermore, there has been some work on representation learning for knowledge graphs (Alam et al., 2022; Wang et al., 2021). However, this line of work typically focuses on downstream tasks such as link prediction, which are very different from the tasks we are focusing on.

Table Representation Learning Workshop at NeurIPS 2022

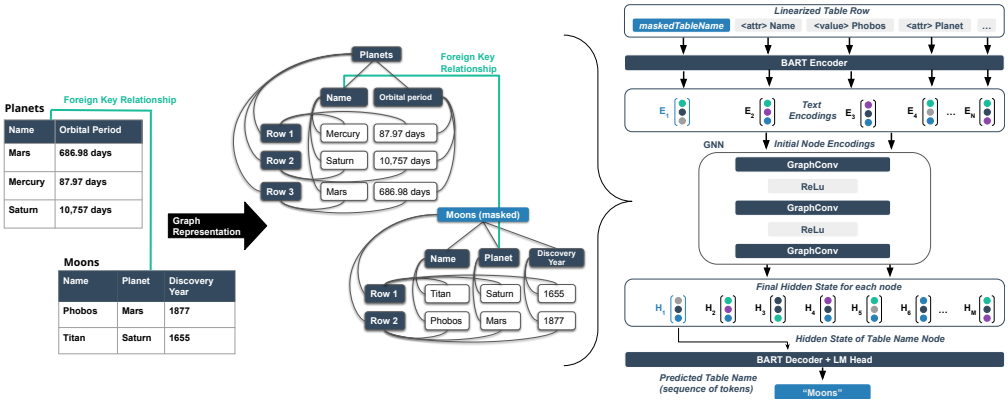

Figure 1: Relational Foundation Models combine Language Models (LMs) with Graph Neural Networks (GNNs). An example of the graph representation is shown on the left-hand side and the architecture is shown on the right-hand side.

Contributions. In this paper, we thus introduce our vision of relational representation learning that can not only learn from the full relational structure instead of only single tables, but also scales to large databases. The main idea of the new model architecture to tackle these issues is to combine language models (LMs) with graph neural networks (GNNs), which is a combination that has already successfully been used in other domains (Ioannidis et al., 2022). As the core contribution of this paper, we discuss the design of such a model for relational representation learning. The main idea behind the model is that we use LMs to encode individual table rows and their schema to model the relational structure within tables (e.g., which rows belong to the same table) as well as structure across tables (e.g., relationships between rows).
Moreover, as a second contribution, we show initial highly promising results of using our new model architecture for three representative tasks on two data sets, where we **outperform existing single-table models by more than** $2\times$ **in accuracy** in the best case. Overall, based on our initial results, we believe that our approach for learning the representation of relational data can actually lead us towards **foundation models for relational data** which exist today only for text & images.

2 Overall Vision

Foundation models like GPT-3 (Brown et al., 2020) for text are pre-trained in a self-supervised manner on very large corpora and can be adapted to solve downstream tasks with comparatively little effort. However, for relational data, a foundation model still needs to be developed.

Relational Foundation Models. The vision that we propose in this paper is to pre-train a high-capacity model on a large corpus of different relational databases. The data is supposed to cover various sizes of databases from a large variety of domains. This allows for a model that generalizes out-of-the-box to previously unseen databases. As we have shown in our initial experiments in Section 4, such a relational foundation model has the potential to greatly facilitate the automation of data engineering tasks where a small amount of annotated training data suffices to adapt the model to tasks like schema matching or entity resolution.
To enable such a pre-trained relational model that can learn from a wide range of different relational databases in a self-supervised manner, we propose a new model architecture as shown in Figure 1. As discussed before, our proposed model architecture for relational representation learning is a combination of language models (LMs) with graph neural networks (GNNs). However, blindly combining LMs with GNNs will not lead to a foundation model for relational data that can provide high performance and scale to large databases.

Core Design Principles. In the following, we present the core design principles of our model architecture to achieve scalability and high accuracy of learned relational representations.
Design for Scale. Existing approaches such as TURL (Deng et al., 2020) are limited in their scalability since they linearize the full table (i.e., all rows) as input to a language model. To address this issue, in our model architecture we only linearize individual rows together with their schema information (i.e., column names and table names) which we use as input for an LM encoder (i.e., BART (Lewis et al., 2020) in our case) as shown in Figure 1 (right). The row-wise embeddings are then combined by the GNN to learn a relational embedding across rows and tables as we discuss next. The representations learned by our current model implementation can be used for enabling various downstream tasks by

using a task-specific LM-head as shown in Figure 1 (right). Similar to our model, RPT also encodes table data row-wise, but as a major difference RPT does not learn higher-level representations on the table-level or even across tables, and thus the representations lead to much lower accuracy for various tasks, as we show in our initial evaluation in Section 4.

Design for Expressiveness. In order to learn a relational representation on the level of tables and across tables, we model the structure of the relational data as a graph that we use as basis for the GNN. Modeling relational data as graph for a GNN comes natural in many places and various different approaches exist. In fact, our graph structure as shown in Figure 1 (right) is inspired by Cvitkovic (2020). However, it is far from obvious how LMs and GNNs should be best combined to learn a deep contextualized representation of relational data and how such models can be efficiently trained. In the following, we hence discuss the current state of our implementation as well as open challenges.

3 Current State and Open Challenges

In the following, we discuss the current state of our model implementation as well as open challenges.

Pre-training Procedure. With our model, we have the ability to learn representations by taking the full relational structure into consideration. For example, to learn a representation of the table *Moons* in Figure 1, all columns of that table (with the names and cell values) as well as neighboring tables are considered. As a pre-training task to learn a relational representation, we use masked value reconstruction on the different levels of the model; i.e., we use masked cell values, masked column names, and masked table names. In our current implementation, we first fine-tune a pre-trained LM on individual table rows and then freeze its weights when pre-training the representations (i.e., for cell values, columns and tables) using the GNN, which propagates the information across rows and tables based on message passing. Another alternative would be to pre-train the entire model including the LM end-to-end, which is clearly more resource intensive but might yield even better accuracy.

Datasets for Pre-training. For pre-training, a large training corpus of relational databases with different data characteristics is needed. Unfortunately, the available large table corpora such as *gitTables* (Hulsebos et al., 2021) contain only single (web-)tables and thus lack relationships between the tables. As such, constructing a corpus of relational databases with connected tables is an open challenge. One starting point is the *Relational Learning Repository* (Motl and Schulte, 2015). Unfortunately, this repository only has a limited number of databases and many databases are also rather small in size. As such, another promising direction is to use *wikiData* (Vrandecic and Krötzsch, 2014) to automatically extract a corpus of relational databases from separate Wiki domains with tables that are connected.

Support for Wide Tables. As discussed before, our model architecture has the ability to learn representations from databases with a large number of rows. However, for wide tables with a high number of columns, as they can be found in real-world databases as well, the serialized representation of a table row might still exceed the maximum number of tokens that an LM can process (i.e. 1024 tokens for BART). For supporting wide tables in our model architecture, several options exist, such as truncating the tables or vertically splitting them into several smaller (connected) tables. However, how to split tables in an optimal manner without loosing context in the row encoding is not clear.

Efficient Learning for Large Databases. Relational database tables can often be very large in size and thus also lead to large graphs. However, processing large graphs with standard GNN techniques is costly regarding time and resources. One obvious way to tackle this is to use only smaller table samples or use only n-hop neighboring tables, which however, means that we lose context for learning relational representations. An alternative, more promising direction is to train a model on the full graph with more efficient training procedures, such as specialized approaches for large graphs, e.g., GraphSAGE (Hamilton et al., 2017). Another approach could be to express relational data as acyclic directed graphs by performing a breadth-first traversal (BFT) of the data. For such acyclic directed graphs, recent work (Thost and Chen, 2021) has shown that a single-pass training procedure can be used that provides much more efficient training for large graphs, in contrast to using multiple rounds of message passing as common in GNNs.

Featurization of Relational Data. In our current prototype, we only use the row-wise embeddings of the LM to initialize the nodes of the GNN. However, in the future we plan to also include some statistical features as input to the graph nodes, e.g., min/max values or information on distinct values of columns that could further improve the model performance (e.g., for numerical columns). Moreover, including some statistical features will potentially allow a foundation relational model to be used for a much wider set of tasks on relational data, such as learned approximate query answering (Hilprecht et al., 2020; Ma and Triantafillou, 2019) or even for learning database internal tasks, such as cardinality estimations (Yang et al., 2019) that today rely on distinct model architectures.

Table 1: Results of our initial experiments on three different tasks. The RPT Baseline is our re-implementation of RPT by Tang et al. (2021), since the code of RPT was not available.

Dataset	Approach	Accuracy for mask reconstruction [%]		
		Task 1: Missing Values	Task 2: Column Name Detection	Task 3: Table Name Detection
wikiTables	$\text{BART}_{table\ (\text{encoder\&decoder})}$ (RPT Baseline)	20.75	66.88	36.99
	$\text{BART}_{table\ (\text{encoder\&decoder})}$ + GNN (Ours)	**46.15**	**83.91**	**37.85**
	$\text{BART}_{text\ (\text{encoder})}$ + $\text{BART}_{table\ (\text{decoder})}$ + GNN (Ablation Study)	37.24	63.77	37.24
	$\text{BART}_{text\ (\text{encoder\&decoder})}$+ GNN (Ablation Study)	24.25	20.18	3.13
gitTables	$\text{BART}_{table\ (\text{encoder\&decoder})}$ (RPT Baseline)	21.65	46.63	**59.71**
	$\text{BART}_{table\ (\text{encoder\&decoder})}$ + GNN (Ours)	**52.63**	**90.04**	52.63
	$\text{BART}_{text\ (\text{encoder})}$+ $\text{BART}_{table\ (\text{decoder})}$ + GNN (Ablation Study)	49.00	73.15	39.54
	$\text{BART}_{text\ (\text{encoder\&decoder})}$ + GNN (Ablation Study)	35.32	24.45	11.78

4 Initial Evaluation

In this section, we present initial very promising results of our approach. As a main result, we show that including the full relational structure is beneficial for learning the relational structure.

Pre-training of Model. For pre-training the model, as mentioned before, we train the LM (BART in our case) and the GNN separately. For training BART on table data, we serialize table rows as discussed in Yin et al. (2020); Tang et al. (2021) and train BART to reconstruct masked table names, column names and cell values per row. As GNN, we use a graph convolutional neural network (Kipf and Welling, 2017). For training the GNN, we use the BART encoder to compute node embeddings and the BART decoder to convert the representation of a masked node in the GNN back into natural language text. We calculate the cross entropy loss between the decoder output and the true label and adapt the gradients of the GNN.

Training Data. The goal of our experiment is to demonstrate that incorporating the full relational structure can improve tabular representation learning. Since to the best of our knowledge there are no baseline approaches available, which support multi-table relational datasets, we resort to single table datasets in this experiment, in particular *gitTables* (Hulsebos et al., 2021) and *wikiTables* (Bhagavatula et al., 2015). However, interestingly, encoding the data of single tables as graphs and thus incorporating the relational structure more efficiently already leads to significant improvements compared to RPT, which uses only a LM. For training, we take a subset of 10,000 tables of each corpus and split it into 70/20/10 for train, validation and test set. The checkpoint with the best accuracy on the validation set is used for evaluation and we report results as an average of three runs.

Experiments. In our experiments, we use a pre-trained BART model from the HuggingFace library (Wolf et al., 2020) which we denote as BART_{text}. For our model, we further fine-tune this model on each of the two tabular datasets as discussed before. We denote this model as BART_{table} and use it as our encoder to create initial embeddings for the GNN. In addition, we also show two variants of our model as an ablation study, where we use the pre-trained model BART_{text} only as encoder or both encoder/decoder, without fine-tuning.

Initial Results. The results of three data engineering tasks that have been used in the literature (Deng et al., 2020) are shown in Table 1. Our results show, that we achieve considerably higher performance when compared to the RPT baseline, especially for the tasks of missing value prediction and column name detection. RPT outperforms our approach only on the table name detection task on *gitTables* data, where we had to use the filenames as table names, which are often not very expressive. Hence we speculate that learning from the full table structure does not help our approach. Overall, our results demonstrate, that there is value in taking advantage of the relational structure in form of a graph, as it allows the model to incorporate more context. Moreover, our ablation studies show, that it is valuable to fine-tune the LM to tabular data, as using a LM pre-trained only on text results in much lower performance.

5 The Road Ahead

In this paper, we have described our vision of using representation learning for relational databases. We discussed opportunities and challenges of developing foundation models for relational data. Our initial results have shown that representing tables as a combination of GNNs and LMs leads to learning more comprehensive representations. Overall, this a an important but only first step towards enabling the vision of foundation relational models and many open research challenges need to be addressed to enable the full vision.

References

Mirza Mohtashim Alam, Md. Rashad Al Hasan Rony, Mojtaba Nayyeri, Karishma Mohiuddin, M. S. T. Mahfuja Akter, Sahar Vahdati, and Jens Lehmann. 2022. Language Model Guided Knowledge Graph Embeddings. *IEEE Access* 10 (2022), 76008–76020. https://doi.org/10.1109/ACCESS.2022.3191666

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: Entity Linking in Web Tables. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 9366)*, Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab (Eds.). Springer, 425–441. https://doi.org/10.1007/978-3-319-25007-6_25

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 1335–1349. https://doi.org/10.1145/3318464.3389742

Milan Cvitkovic. 2020. Supervised Learning on Relational Databases with Graph Neural Networks. *CoRR* abs/2002.02046 (2020). arXiv:2002.02046 https://arxiv.org/abs/2002.02046

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proc. VLDB Endow.* 14, 3 (2020), 307–319. https://doi.org/10.5555/3430915.3442430

Raul Castro Fernandez and Samuel Madden. 2019. Termite: a system for tunneling through heterogeneous data. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management, aiDM@SIGMOD 2019, Amsterdam, The Netherlands, July 5, 2019*, Rajesh Bordawekar and Oded Shmueli (Eds.). ACM, 7:1–7:8. https://doi.org/10.1145/3329859.3329877

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 1025–1035.

Benjamin Hilprecht, Andreas Schmidt, Moritz Kulessa, Alejandro Molina, Kristian Kersting, and Carsten Binnig. 2020. DeepDB: Learn from Data, not from Queries! *Proc. VLDB Endow.* 13, 7 (2020), 992–1005. https://doi.org/10.14778/3384345.3384349

Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2021. GitTables: A Large-Scale Corpus of Relational Tables. *CoRR* abs/2106.07258 (2021). arXiv:2106.07258 https://arxiv.org/abs/2106.07258

Vassilis N. Ioannidis, Xiang Song, Da Zheng, Houyu Zhang, Jun Ma, Yi Xu, Belinda Zeng, Trishul Chilimbi, and George Karypis. 2022. Efficient and effective training of language and graph neural network models. *CoRR* abs/2206.10781 (2022). https://doi.org/10.48550/arXiv.2206.10781 arXiv:2206.10781

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

Jens Krüger, Changkyu Kim, Martin Grund, Nadathur Satish, David Schwalb, Jatin Chhugani, Hasso Plattner, Pradeep Dubey, and Alexander Zeier. 2011. Fast Updates on Read-Optimized Databases Using Multi-Core CPUs. *Proc. VLDB Endow.* 5, 1 (2011), 61–72. https://doi.org/10.14778/2047485.2047491

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Qingzhi Ma and Peter Triantafillou. 2019. DBEst: Revisiting Approximate Query Processing Engines with Machine Learning Models. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 1553–1570. https://doi.org/10.1145/3299869.3324958

Jan Motl and Oliver Schulte. 2015. The CTU Prague Relational Learning Repository. *CoRR* abs/1511.03086 (2015). arXiv:1511.03086 http://arxiv.org/abs/1511.03086

Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? https://doi.org/10.48550/ARXIV.2205.09911

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. https://doi.org/10.14778/3457390.3457391

Veronika Thost and Jie Chen. 2021. Directed Acyclic Graph Neural Networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=JbuYF437WB6

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. https://doi.org/10.1145/2629489

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Trans. Assoc. Comput. Linguistics* 9 (2021), 176–194. https://doi.org/10.1162/tacl_a_00360

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*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Zongheng Yang, Eric Liang, Amog Kamsetty, Chenggang Wu, Yan Duan, Xi Chen, Pieter Abbeel, Joseph M. Hellerstein, Sanjay Krishnan, and Ion Stoica. 2019. Deep Unsupervised Cardinality Estimation. *Proc. VLDB Endow.* 13, 3 (2019), 279–292. https://doi.org/10.14778/3368289.3368294

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 8413–8426. https://doi.org/10.18653/v1/2020.acl-main.745