WikiDBs: A Corpus Of Relational Databases From Wikidata

Liane Vogel[1,*], Carsten Binnig[1,2]

[1]Technical University of Darmstadt, Darmstadt, Germany
[2]DFKI Darmstadt, Germany

Abstract

In recent years, deep learning on tabular data, also known as tabular representation learning, has gained growing interest. However, representation learning for relational databases with *multiple* tables is still an under-explored area, which might be due to the lack of openly available resources. Therefore, we introduce WikiDBs, a novel open-source corpus of **10,000 relational databases**. Each database consists of multiple tables that are connected by foreign keys. The dataset is based on Wikidata and aims to follow the characteristics of real-world databases. In this paper, we describe the dataset and the method for creating it. We also conduct preliminary experiments on the tasks of imputing missing values and predicting column and table names in the databases.

1. Introduction

The Importance of Representation Learning. While text and images often dominate the field of representation learning, considerable progress has recently also been made on other modalities such as tabular data [1, 2]. This is important since a non-negligible amount of data is expressed in tabular form, in particular enterprise data [3]. For individual tables, several approaches have been developed to solve downstream tasks such as entity matching or missing value imputation. Several large-scale datasets, such as GitTables [4] and WikiTables [5], provide the necessary training data, as data availability is essential for the development of proficient deep learning models.

Missing Large Corpora for Relational Databases. For relational databases with multiple tables that are linked with foreign keys, however, there is a lack of both large openly available training data and deep neural network architectures that can incorporate the context of multiple related tables. However, collecting large corpora of relational data is non trivial. Due to the sensitivity of data stored in relational databases, real-world enterprise databases are typically kept private and are not accessible to the representation learning community, resulting in a lack of openly available databases.

The need for Real-World Data. As a consequence, in the field of database research, it is common to use synthetic databases such as the datasets in the TPC benchmarks [6, 7]. This may be sufficient for testing database-internals, but for representation learning on relational databases, which requires a large number of different databases, the existing benchmarks are too few and not diverse enough in terms of the domains they cover. In addition, automatic data generation is not a valid option, as current methods are only able to generate numerical and categorical data, but not meaningful text and context as contained in real-world databases. According to [8], a significant part of the data in databases is saved as text, so in order to have realistic training data, it is important to use databases that contain not only numeric and categorical, but also textual data.

Towards a new Corpus of Relational Databases. We aim to support research on representation learning for relational data by creating a new, large-scale resource for tabular representation learning on relational databases. Hereby, our goal is to have realistic data, that is not synthetically generated. While a few real-world relational databases exist that are openly available such as the Internet Movie Database (IMDb) or the MIMIC database [9], no large corpus containing many relational databases exists. Therefore, we present an approach that uses the Wikidata knowledge base [10] as the basis for deriving a large corpus of relational databases. Along with this paper, we are releasing a new, open-access dataset called WikiDBs — a corpus of 10,000 relational databases extracted from Wikidata covering a wide spectrum of diverse domains.

Initial Results using the Corpus. In this paper, we compare the characteristics of our corpus to statistics available for real-world relational databases to justify the design of our corpus. Furthermore, to showcase that the corpus can be used to learn representations that are informed by multiple tables in a relational database, we follow an approach presented in [11]. In this work, we introduced the vision of new models for representation learning on relational databases. Here, we demonstrate first experiments of such a model which is trained on our new WikiDBs dataset.

Contributions of this Work. To summarize, this paper makes the following contributions: (1) We introduce a novel method of extracting multi-table relational

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — TaDA'23: Tabular Data Analysis Workshop, August 28 - September 1, 2023, Vancouver, Canada
*Corresponding author.
liane.vogel@cs.tu-darmstadt.de (L. Vogel);
carsten.binnig@cs.tu-darmstadt.de (C. Binnig)
0000-0001-9768-8873 (L. Vogel); 0000-0002-2744-7836 (C. Binnig)

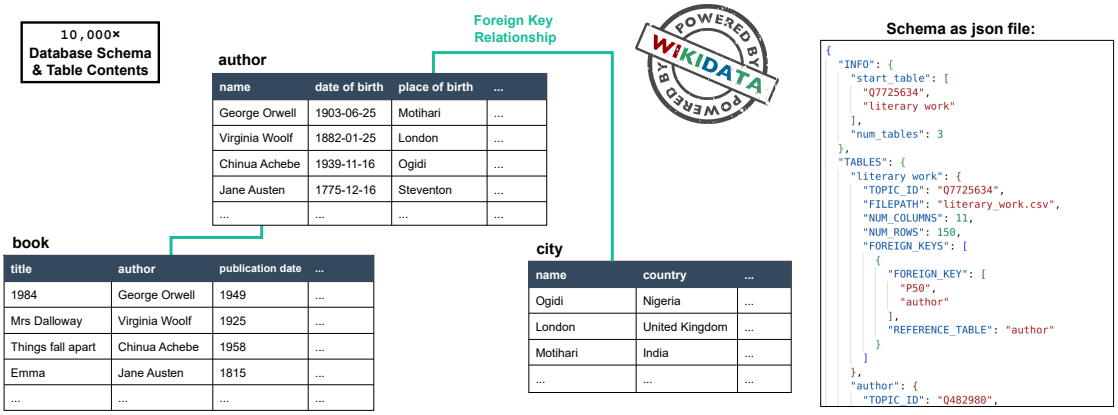

Figure 1: We release **WikiDBs**, a dataset of 10,000 databases based on data from Wikidata. The figure shows one of the database schemas with examples of the tables' content (left) and how the schema is stored as a JSON file analogous to the GitSchemas dataset [12] (right).

databases from Wikidata. (2) We release a first large scale corpus of relational data[1] and derive important statistics which we compare to available characteristics of real-world relational databases. (3) We show first experimental results on our dataset for the tasks of missing value imputation, table and column name prediction.

2. The WikiDBs Dataset

2.1. Design Principles

For our WikiDBs dataset, we want the characteristics to reflect the properties of real-world databases. As enterprises do not share the statistics of their databases, we have to rely on the characteristics of available public resources and model our dataset in a similar way. In Table 1, we have collected characteristics of existing public resources, such as the number of tables in a database, and the average number of columns and rows per table.

For deriving statistics, we found only two existing collections of relational databases, the Relational Learning Repository from CTU Prague [13] (which also includes TPC-H and IMDb) and the SQLShare [14] repository. However, all these repositories include only a small number of relational databases. Therefore, we also include the statistics of the significantly larger datasets GitSchemas [12] — which only contains schema information — and GitTables [4] — a corpus of individual tables. For the distribution of how many tables we include per database, we follow the distribution of the GitSchemas dataset, which is based on a large number of database schemas found in public git repositories. We include on average a higher number of columns per table as e.g. the CTU Paragues dataset or GitSchemas, because real-world enterprise data also often includes a large number of columns, e.g. 18.7 on average as reported in the SQLShare corpus [14].

For our corpus, we store the schema information which includes the table structure and the foreign keys. For the schema information, we use the same format that is used by the GitSchemas dataset (shown in Figure 1, right). Furthermore, the individual table data is made available in the CSV-format.

2.2. Analysis and Statistics

Next, we analyze the resulting statistics of the derived corpus which is published with this paper. Overall, as mentioned before, our dataset consists of 10,000 databases that each have between two and nine tables which are connected via foreign keys. The statistics of WikiDBs are compared to those of existing open resources in Table 1. In total, our dataset contains 42,472 tables, the median number of tables per database is 4. On average, each table has 17.9 columns and 46.3 rows. The distribution of the number of tables per database is visualized in Figure 2.

2.3. Methodology of Construction

In this section we describe the procedure how we derive relational databases based on Wikidata.

Wikidata Dataformat. The data in Wikidata is stored in a document-oriented database, where documents represent items that are instances of different concepts, such as *artists* or *paintings*. In this way, concepts closely resemble the notion of tables.

Every item in Wikidata is associated with a unique identifier, the so-called *QID*. The item representing the book *1984* by George Orwell for example has the id *Q208460*. Properties of items are stored in form of key-value pairs, where property names are saved with their

[1] https://wikidbs.github.io

Table 1
Characteristics of existing resources compared to our new dataset. We report the median number of tables per database, as well as the average number of columns and rows. GitSchemas [12] does not contain the content (=rows) of the databases and GitTables [4] consists of single tables, not databases.

	includes schema	includes table content	#DBs	#Tables	#Tables per DB	#Colums	#Rows
					Median	Avg.	Avg.
CTU Prague [13]	✓	✓	83	813	5	6.0	4.8k
SQLShare [14]	✓	✓	64	3.9k	4	18.7	11k
GitSchemas [12]	✓	✗	156k	1.2M	4	5.7	-
GitTables [4]	✗	✓	-	1M	1	12.0	142
WikiDBs (ours)	✓	✓	**10k**	**42.5k**	**4**	**17.9**	**46**

corresponding value. Most important are properties (e.g. the *publication date (P577)*) that resemble attributes of a table row. Moreover, properties also include other information such as the related concept of an item; e.g. the book *1984* has the property *instance of (P31) literary work (Q7725634)*.

Creation of a Table. The creation of a relational table from Wikidata is thus made possible by the *instance of* (or also the *part of* or *subclass of*) relations in Wikidata. The information that the book *1984* is an instance of *literary work* allows us to search Wikidata for all other items that are also tagged with the information that they are an instance of *literary work*.

A challenge in Wikidata is that every item (e.g., each book) might use a different set of properties. For example, for some books the year the book was published is available, while for others it is not. For constructing tables, we use the union of all properties. If a value is missing, we store a NULL-value in the table row. To avoid constructing tables with highly sparse columns, we prune columns of a table where the fraction of NULL-values is beyond a configurable threshold[2].

Creation of a Database. For each created table, some columns contain references to concepts that are also saved as items in Wikidata. We use those columns that contain Wikidata items to build further tables for the database. For constructing relational databases, we randomly select a concept in Wikidata as a starting point and then traverse relationships to other tables randomly. For example, the table of *literary work* contains a column *author* which is a reference to another item, which allows us to build an additional table of *authors* (linked via a foreign key to the table *literary work*) where each row contains information on an author and columns contain e.g. their date and place of birth or nationality (Figure 1).

Implementation Details. For constructing our corpus, we use the Wikidata JSON dump [15] as a starting point for creating the dataset. To enable efficient querying of data in the dump, we additionally build up a lookup structure which maps concepts (i.e., tables) to potential items (i.e., rows). The values of each item correspond to the rows of the tables, the properties of the values form the column headers. The lookup structure allows us to quickly navigate the content in the dump and extract data for individual tables

[2]For the corpus released with this paper, we prune columns with more than 20% NULL-values.

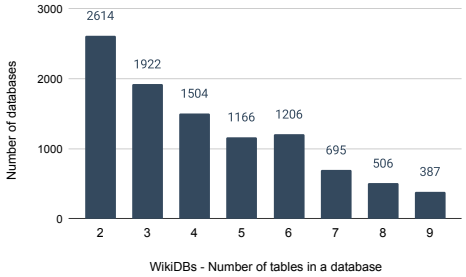

Figure 2: Distribution of the number of tables over the dataset. The distribution is modeled analogous to the GitSchemas [12] dataset, e.g. 26% of the databases contain 2 tables.

2.4. Discussion

We clearly see this work and the corpus released with this paper only as a starting point to foster further research. While this is the very first large scale corpus of relational databases, we believe that more work is necessary to extend the corpus. First, at the moment we provide tables of sizes which closely resemble the sizes of tables found in repositories such as on GitHub. However, real-world enterprise databases often contain also a few very large tables (e.g., the orders table of an online shop). For Wikidata, we found approximately 20 concepts such as *scholarly article*, *galaxy* or *protein* that have more than 400k items, enabling the creation of very large tables and databases. Furthermore, with our repository we focus on English-language content in the first version but our method allows to easily create databases in other languages included in Wikidata.

Table 2
Results of our initial experiments on three different tasks on our new WikiDBs dataset. The RPT Baseline is our re-implemenatation of RPT [16], since the code of RPT was not available.

Approach	Data	Accuracy for mask reconstruction [%]		
		Task 1: Missing Values	Task 2: Column Name Detection	Task 3: Table Name Detection
$BART_{table}$ (RPT Baseline)	single tables	24.14	48.98	48.80
$BART_{table}$ + GNN (ours)	databases	30.22	69.37	50.08

Finally, we hope that the corpus fosters more research on models for table representation that can take data from multiple connected tables into account. In the following, we show the results of an early version of such a model that is enabled by the WikiDBs corpus.

3. Experiments

In this section, we conduct initial experiments on our new WikiDBs dataset. We present results for three different tasks, namely predicting missing values, column names and table names. We model all these tasks as generative tasks rather than classification tasks in order to be able to work with unseen data. We apply the architecture introduced in [11] that is a combination of language models (LMs) and graph neural networks (GNNs). Similar to [11], we compare our model to RPT [16] as a baseline.

Pre-Training Procedure. Following [11], we train the language model BART [17] and the GNN separately. We split the 10,000 databases from our dataset into 80/10/10 percent for training, validation and testing. First, we fine-tune a pre-trained BART model from the Huggingface library [18] on single tables from our dataset for 250 epochs with an initial learning rate of $10e-4$ and a cosine annealing schedule to reconstruct masked table names, column names and cell values. Next, for training the GNN on the databases, we use the fine-tuned BART encoder to compute node embeddings for a database and the BART decoder to convert the representation of a masked node in the GNN back into natural language text. In our experiments, we limit a database to using a table and its direct neighbors. We train the GNN for 500 epochs. The checkpoint with the best accuracy on the validation set is used for evaluation and we report the results as an average of three runs.

Initial Results. The results of our initial experiments on the WikiDBs corpus are shown in Table 2. Compared to the RPT [16] baseline that is only able to work on single tables, our model achieves a higher performance for all three tasks. Incorporating the context of multiple tables of the databases increases the F1 score especially for the task of column name detection, from 48.98% for the BART model to 69.37% for our model.

4. Related Work

In the following, we summarize related work grouped by different directions.

Single-Table Repositories. So far, tabular representation learning mostly focuses on learning representations of single tables. Commonly used corpora are for example GitTables [4], WikiTables [5], the Dresden Web Table Corpus (DWTC) [19] or the WDC corpora [20].

Multi-Table Repositories. In order to support machine learning on multi-table relational data, [13] published the CTU Prague Relational Learning Repository in 2015. Currently, there are 83 databases included. The SQLShare corpus [14] is a query workload dataset which includes 64 databases collected from real-world users (mainly researchers and scientists) from the webservice SQLShare. Both repositories are thus much smaller than corpora with data for single tables that are commonly used for table representation learning. Finally, the GitSchemas [12] repository contains 50k database schemas based on SQL files from public GitHub repositories. The information thus provides highly relevant insight into real-world databases. However, the repository lacks the content of the databases.

Datasets based on Wikidata. For the SemTab challenge [21], where tabular data is matched to knowledge graphs, tables were built using data from Wikidata. Furthermore, Wikidata has been used to build datasets for named entity classification [22] and named entity disambiguation [23], as well as complex sequential question answering [24]. Moreover, [25] verbalize knowledge graph triples from Wikidata, and [26] create alignments between Wikidata triples and Wikipedia abstracts.

5. Conclusion & Future Work

To support representation learning on databases we introduced our new dataset WikiDBs that is based on data from Wikidata and released a first corpus with 10,000 databases. In future, we plan to extend the dataset and look into opportunities to leverage the corpus for new model architectures or for fine-tuning large language models such as GPT-based [27] models on table data.

Acknowledgments

We thank Till Döhmen and Madelon Hulsebos for generously providing the table statistics from their GitSchemas dataset. This work has been supported by the BMBF and the state of Hesse as part of the NHR Program and the BMBF project KompAKI (grant number 02L19C150), as well as the HMWK cluster project 3AI. Finally, we want to thank hessian.AI, and DFKI Darmstadt for their support.

References

[1] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, TURL: table understanding through representation learning, Proc. VLDB Endow. 14 (2020) 307–319. URL: http://www.vldb.org/pvldb/vol14/p307-deng.pdf.

[2] G. Badaro, M. Saeed, P. Papotti, Transformers for Tabular Data Representation: A Survey of Models and Applications, Transactions of the Association for Computational Linguistics 11 (2023) 227–249. URL: https://doi.org/10.1162/tacl_a_00544.

[3] J. Cahoon, A. Savelieva, A. C. Mueller, A. Floratou, C. Curino, H. Patel, J. Henkel, M. Weimer, N. Gustafsson, R. Wydrowski, R. Batoukov, S. Deep, V. Emani, The need for tabular representation learning: An industry perspective, in: NeurIPS 2022 First Table Representation Workshop, 2022. URL: https://openreview.net/forum?id=jk4B84qmlXJ.

[4] M. Hulsebos, Ç. Demiralp, P. Groth, Gittables: A large-scale corpus of relational tables, arXiv preprint arXiv:2106.07258 (2021). URL: https://arxiv.org/abs/2106.07258.

[5] C. S. Bhagavatula, T. Noraset, D. Downey, 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, volume 9366 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 425–441. URL: https://doi.org/10.1007/978-3-319-25007-6_25.

[6] K. Huppler, The art of building a good benchmark, in: Performance Evaluation and Benchmarking, First TPC Technology Conference, TPCTC 2009, Lyon, France, August 24-28, 2009, Revised Selected Papers, volume 5895 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 18–30. URL: https://doi.org/10.1007/978-3-642-10424-4_3.

[7] M. Poess, B. Smith, L. Kollar, P. Larson, Tpc-ds, taking decision support benchmarking to the next level, in: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD '02, Association for Computing Machinery, New York, NY, USA, 2002, p. 582–587. URL: https://doi.org/10.1145/564691.564759.

[8] A. Vogelsgesang, M. Haubenschild, J. Finis, A. Kemper, V. Leis, T. Mühlbauer, T. Neumann, M. Then, Get real: How benchmarks fail to represent the real world, in: Proceedings of the 7th International Workshop on Testing Database Systems, DBTest@SIGMOD 2018, Houston, TX, USA, June 15, 2018, ACM, 2018, pp. 1:1–1:6. URL: https://doi.org/10.1145/3209950.3209952.

[9] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R. G. Mark, Mimic-iii, a freely accessible critical care database, Scientific data 3 (2016) 1–9. URL: https://doi.org/10.1038/sdata.2016.35.

[10] Wikidata, https://www.wikidata.org, .

[11] L. Vogel, B. Hilprecht, C. Binnig, Towards foundation models for relational databases [vision paper], NeurIPS 2022 First Table Representation Workshop (2022). URL: https://openreview.net/forum?id=s1KlNOQq71_.

[12] T. Döhmen, M. Hulsebos, C. Beecks, S. Schelter, Gitschemas: A dataset for automating relational data preparation tasks, in: 38th IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2022, Kuala Lumpur, Malaysia, May 9, 2022, IEEE, 2022, pp. 74–78. URL: https://doi.org/10.1109/ICDEW55742.2022.00016.

[13] J. Motl, O. Schulte, The CTU prague relational learning repository, CoRR abs/1511.03086 (2015). URL: http://arxiv.org/abs/1511.03086. arXiv:1511.03086.

[14] S. Jain, D. Moritz, D. Halperin, B. Howe, E. Lazowska, Sqlshare: Results from a multi-year sql-as-a-service experiment, in: Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016, ACM, 2016, pp. 281–293. URL: https://doi.org/10.1145/2882903.2882957.

[15] Downloaded the 'latest-all.json.gz' dump on February 22, 2023 from https://dumps.wikimedia.org/wikidatawiki/entities/.

[16] N. Tang, J. Fan, F. Li, J. Tu, X. Du, G. Li, S. Madden, M. Ouzzani, RPT: relational pre-trained transformer is almost all you need towards democratizing data preparation, Proc. VLDB Endow. 14 (2021) 1254–1261. URL: http://www.vldb.org/pvldb/vol14/p1254-tang.pdf.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, 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, Association for Computational Linguistics, 2020, pp. 7871–7880. URL: https://doi.org/10.18653/v1/2020.acl-main.703.

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, 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, Association for Computational Linguistics, 2020, pp. 38–45. URL: https://doi.org/10.18653/v1/2020.emnlp-demos.6.

[19] J. Eberius, M. Thiele, K. Braunschweig, W. Lehner, Top-k entity augmentation using consistent set covering, SSDBM '15, 2015. doi:10.1145/2791347.2791353.

[20] O. Lehmberg, D. Ritze, R. Meusel, C. Bizer, A large public corpus of web tables containing time and context metadata, in: Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume, ACM, 2016, pp. 75–76. URL: https://doi.org/10.1145/2872518.2889386.

[21] E. Jiménez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems, in: The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings, volume 12123 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 514–530. URL: https://doi.org/10.1007/978-3-030-49461-2_30.

[22] J. Geiß, A. Spitz, M. Gertz, Neckar: A named entity classifier for wikidata, in: Language Technologies for the Challenges of the Digital Age - 27th International Conference, GSCL 2017, Berlin, Germany, September 13-14, 2017, Proceedings, volume 10713 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 115–129. URL: https://doi.org/10.1007/978-3-319-73706-5_10.

[23] A. Cetoli, M. Akbari, S. Bragaglia, A. D. O'Harney, M. Sloan, Named entity disambiguation using deep learning on graphs, CoRR abs/1810.09164 (2018). URL: http://arxiv.org/abs/1810.09164. arXiv:1810.09164.

[24] A. Saha, V. Pahuja, M. M. Khapra, K. Sankaranarayanan, S. Chandar, Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 705–713. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17181.

[25] G. Amaral, O. Rodrigues, E. Simperl, WDV: A broad data verbalisation dataset built from wikidata, in: The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings, volume 13489 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 556–574. URL: https://doi.org/10.1007/978-3-031-19433-7_32.

[26] H. ElSahar, P. Vougiouklis, A. Remaci, C. Gravier, J. S. Hare, F. Laforest, E. Simperl, T-rex: A large scale alignment of natural language with knowledge base triples, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018, European Language Resources Association (ELRA), 2018. URL: http://www.lrec-conf.org/proceedings/lrec2018/summaries/632.html.

[27] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, 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, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.