Chorus: Foundation Models for Unified Data Discovery and Exploration

Moe Kayali
University of Washington

Anton Lykov
University of Washington

Ilias Fountalis
RelationalAI

Nikolaos Vasiloglou
RelationalAI

Dan Olteanu
University of Zurich

Dan Suciu
University of Washington

1 Introduction

Data discovery and exploration are major components of the workflow of analysts and data scientists. A survey conducted by the Anaconda data-science platform in 2021 found that analysts spend 40% of their working hours on data loading and cleaning [2]. Even with this colossal effort, 60-70% of data within an enterprise still goes unused for analytics [11], remaining as *dark data* [12, 37].

Recent developments in large language-models (LLMs) have unlocked human-level performance on diverse domain tasks. The discovery that these models can generalize to diverse domain-specific tasks that they have not been trained on [33, 34, 3, 15] has led to emergence of the term *foundation models* [5]. Despite their promise, serious risks have hampered the reception of foundation models. These include: spurious generation (including "hallucination") [13], factual recall limitations [22] and dataset contamination [9].

The goal of this paper is to demonstrate the utility of foundation models to the data discovery and exploration domain while mitigating the aforementioned risks. We select three representative tasks to show the promise of foundation models:1*table-class detection*,2*column-type annotation* and3*join-column prediction*. An outline of our approach is shown in Figure 1a. We call this approach CHORUS.

Prior work has addressed these tasks individually. Landmark approaches like Sherlock [16] trained deep model architectures for a specific task, requiring 100K-1M labeled data points. More recent work such as DoDuo [27] and TaBERT [36] has focused on *representation learning*, learning embeddings for structured data by improving their performance on one or more downstream tasks.

Foundation models allow a substantially different approach: rather than the classical architecture where the outputs of the model are task-specific, the inputs and outputs of the model are natural language text. Training occurs not on tables or data management tasks specifically, but on general text. Performance on domain-specific tasks is solely by generalization. The promise of foundation models for data profiling, wrangling and imputation has been outlined in a recent papers [32, 31, 23, 4].

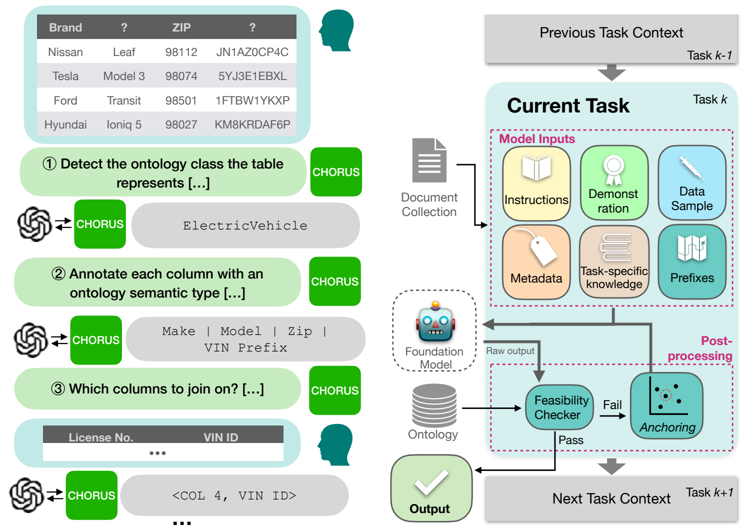
This results in a high degree of flexibility. Novel tasks can be specified in natural text, without need for expensive data collection—task examples, metadata and constraints are all incorporated into the task easily. Another advantage of this approach is a **unified architecture**: tasks can utilize the overall context and previous outputs. For example, in Figure 1a the table class `ElectricVehicle` helps with deducing the outputs `Make`, `Model` in the next task.

Further details on all sections of this paper, including the prompts used, can be found in the full report [19].

2 Background

We assume to be given a *data collection* consisting of a number of relational tables T_1,T_2,.... Each table T_i consists of a number of columns, or attributes, A_1,A_2,... and a number of rows, or tuples, r_1,r_2,... The name of a table T_i is, in general, non-informative, for example it may be simply a sequential ID. The columns may optionally have a name H_1,H_2,... or consist only of values. In addition to the data collection, we are also given a reference ontology of table classes C_1,C_2,..., and a reference ontology of column types τ_1,τ_2,... We consider three tasks of interest on the data collection:

Table Representation Workshop at 37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) An illustration of the three data discovery tasks considered in this work.

(b) CHORUS system architecture.

Definition 2.1 (①Table-class detection). For each table T_i, determine its appropriate class C_j, such that every row $r_1, r_2, \ldots$ represents an instance of the C_j type. We adopt this definition from [20].

Definition 2.2 (②Column-type annotation). For each table T_i, find a mapping from its attributes (columns) $A_1, A_2, \ldots$ to the reference column types $\tau_1, \tau_2, \ldots$, such that each value in A_i is an instance of the τ_i type. See [8, 1].

Definition 2.3 (③Join-column prediction). Assume an *execution log* L, a history of user actions including table joins and their join conditions, which maps many $(T_i, T_j) \to (A_k, A_l)$ where $A_k \in T_i, A_l \in T_j$. Given two tables T and T', with columns $A_1, \ldots$ and $A'_1, \ldots$ respectively, the *join-column prediction* task is to suggest a pair (A_k, A'_l) of columns such that the equality condition $A_k = A'_l$, which can be used to join the the tables, matches with the choice in the execution log L. For more discussion, see [35].

3 Approach

Figure 1b shows the architecture of the system. CHORUS has a unified architecture which runs multiple tasks in the same context, allowing for information flow. Each task is run sequentially, with the output of one task fed as context into future tasks. For each task instance, CHORUS generates a prompt by concatenating six inputs: context, demonstration, data samples, metadata, task-specific knowledge, and prefixes. They form the "Model Inputs" box in Figure 1b. This natural language input is then fed to the foundation model. The output is controlled by a harness: which mitigates for errors of parsability and feasibility.

Model Harness *Constraint checks.* The model may not always output a feasible answer. In this setting we impose three constraints: table types must belong to the ontology classes, column types must belong to the ontology properties and joins must be on existing columns. An output is infeasible if, in particular, it is not parsable or if it violates any of the three constraints. If this occurs, CHORUS performs anchoring.

Anchoring. If the constraints are violated, we do not simply move on to the next task. The risk is of hallucination snowballing [39]: once a foundation model makes a single spurious generation, subsequent outputs are more likely to also be wrong. The model will make mistakes it would otherwise be able to avoid. For example, in Figure 2(a): once nonexistent class `iucnStatus` is suggested, another nonexistent class `animalName` follows. Because we maintain context across tasks, we are particularly vulnerable to this. We call the novel domain-specific mitigation we deploy *anchoring*, shown and explained in Figure 2(b).

4 Experiments

Baselines We considered the following state-of-the-art systems for data exploration: relevant systems include TABERT [36], DODUO [27], Sato [38], TURL [8], TaBBIE [17], Auto-suggest [35], Trifecta Wrangler [30], Paxata, Tableau Prep, and Sherlock. DODUO outperforms TURL and Sherlock on column-type annotation [27], so we select it for evaluation. Sato and Sherlock are similar, with Sato utilizing additional

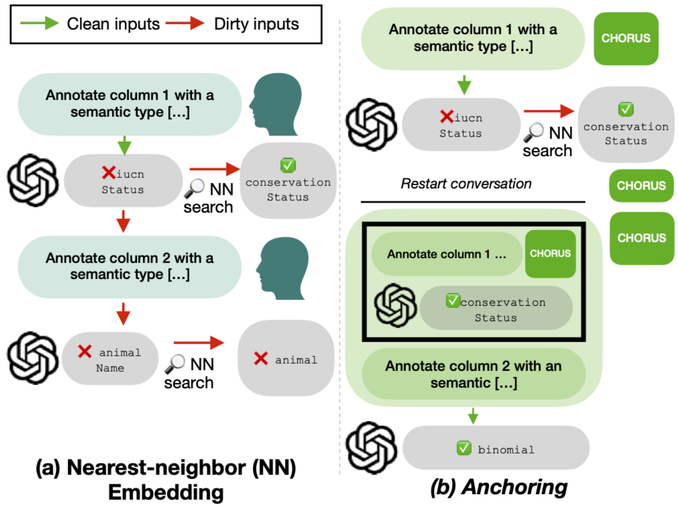


Figure 2: *Anchoring* illustrated. The LLM hallucinates an imagined label, `iucnStatus`. Under the standard approach, this poisons all the upcoming tasks; the nearest-neighbor post-processing cannot recover and outputs the incorrect label `animal`. With anchoring, CHORUS intervenes when the first error is detected. A new conversation is started and a *synthesized (false) history* is provided to the LLM, in which it did not make the mistake. With only clean inputs, LLM is able to correctly answer the next task correctly: `binomial`.

signals not found in our benchmarks, so we evaluate the better-established Sherlock. TaBBIE can embed tables but is not trained on column-type annotation unlike DoDuo and Tabert, so we avoid it for the column-type annotation task. TABERT is a work similar to DoDuo and TURL, but from the NLP community rather than the data management community, so we also test it too. For join-column prediction, Trifacta Wrangler outperforms Paxata and Tableau Prep [35]. Auto-Suggest is reported to outperform Trifacta Wrangler, but is a proprietary research project not released publicly. Thus we select Trifacta Wrangler for testing. In all cases, we use the pretrained embeddings without modification, as provided by the baseline authors. DODUO provides two embedding variants: one trained on the WikiTables dataset and another on VizNet. When using DODUO as a baseline we test against both, labelling them DODUO-WIKI and DODUO-VIZ respectively. We use the GPT-3.5 model [25] for the bulk of experiments.

4.1 Table-class detection

For the first task, ① table-class detection, we tag each table with the DBPedia ontology entry that represents the row-type of the data. The 237 datasets that comprise the T2Dv2 dataset [26] with table-class correspondences available. We compare against the baselines DODUO and TABERT. No approach in the prior work provides out-of-the-box capabilities on this task, so we add a classification layer on top of the pretrained embedding layer using the approach from [20].

Table 1 shows the results. CHORUS improves on the three baselines—DoDuo-Viz, DoDuo-Wiki and TaBERT—on all metrics. F_1 score is improved by 0.169 points, precision by 17.5 percentage points and recall by 15.5 percentage points. Of the baselines, DoDuo-Wiki provides the best F_1 and precision, while TaBERT provides the comparable recall. The best performing models, TaBERT and DoDuo-Wiki are trained on CommonCrawl, a superset of the T2Dv2 benchmark. DoDuo-Viz which is trained on the VizNet, a dataset disjoint from T2Dv2, has the weakest performance. The numbers for TaBERT are in line with prior replications [20], while to the best of our knowledge this is the first benchmarking of DoDuo on this task.

4.2 Column-type annotation

Next, we compare the ability of our system to assign classes to table columns. VIZNET is a collection of tables, extracted by the Sherlock [16] team from the VizNet repository [14] of data visualizations and open datasets. VizNet comprises 31 million datasets in total. We selected 10 mutually exclusive DBPedia.org classes to test. We then used stratified sampling to select 1000 columns of each type. We compare against TaBERT [36], DoDuo [27] and Sherlock [16] on this task. Since Sherlock is designed for column annotation, we use the out-of-the-box model provided by the original team. For TaBERT and DoDuo we adopt a minimal shim to adapt to our ten classes. Table 2 contains the results for the VIZNET dataset. Our FM-based approach improves performance on the measured metrics of F_1-score, precision and recall. The best performing method is Sherlock, narrowly beating DoDuo-VizNet, with a 0.930 F_1 score. If we consider methods which

Table 1: Weighted F_1 scores for *table-class detection* on T2Dv2 dataset. Systems are compared with the expert-annotated classes for each table. The $n=237$ tables each correspond to one of 33 `DBPedia.org` classes.

	F_1-score	Precision	Recall
DoDuo-Viz	0.654	66.8%	68.3%
DoDuo-Wiki	0.757	78.6%	76.9%
TaBERT	0.746	76.3%	76.8%
CHORUS	**0.926**	**96.1%**	**92.4%**

Table 2: Weighted F_1 scores for *column-type annotation* on VIZNET, with $n = 1000$ columns. Systems are compared with the "gold standard" classes for each column. Methods which are also pre-trained on VIZNET are marked with an asterisk $*$.

	F_1-score	Precision	Recall
DoDuo-VizNet$*$	0.900	90.3%	89.9%
Sherlock$*$	**0.930**	**92.2%**	**93.1%**
TaBERT	0.380	38.9%	38.3%
DoDuo-Wiki	0.815	82.6%	81.4%
CHORUS	**0.865**	**90.1%**	**86.7%**

Table 3: F_1 scores, precision and recall for the *join-column prediction* task on $n=300$ tables.

	F_1-score	Precision	Recall
Jaccard	0.575	60.7%	54.7%
Levenshtein	0.718	72.3%	71.3%
Trifacta Wrangler	0.823	82.6%	82.0%
CHORUS	**0.895**	**91.0%**	**88.0%**

Table 4: Weighted F_1 scores for *table-class detection* on T2Dv2 dataset, for different choices of foundation model used by CHORUS. Parameter size in brackets.

	Table-class correctness		
Model choice	F_1-score	Precision	Recall
GPT-3.5 (175B)	0.926	96.1%	92.4%
LLaMA 2 (70B)	0.893	92.2%	86.5%
Vicuna/LLaMA (13B)	0.713	79.2%	64.1%
Vicuna/LLaMA (7B)	0.713	75.3%	67.5%

are not specifically pretrained on VizNet (note, which is also the test set) CHORUS is the best performing on all three metrics. It has comparable F_1 and precision to Sherlock, but 6 percentage points lower recall. Note in particular DoDuo-Wiki, which is the same as DoDuo-Viznet without access to VizNet at pretraining time, has a large regression in performance, suggesting lower generalizability.

4.3 Join-column prediction

Finally, we evaluate our approach's ability to suggest which columns are the correct choice for a join, the join-column prediction task. We use the *GitNotebooks* dataset from [35], a collection of Python notebooks (and their associated relational tables) including which joins the user ran, collected from Github. For this task, we compare with three baselines. Jaccard similarity, J, is the first. Two columns are selected such that $\operatorname{argmax}_{c\in C^T, c'\in C^{T'}} J(c,c')$ where $J(X,Y) = |X\cap Y|/|X\cup Y|$. This is a commonly used approach in the literature [6, 7, 24, 35]. Another baseline is Levenshtein distance [21], which selects the pair of column names with the smallest edit distance between them. The final baseline is Trifacta Wrangler [30], a commercial product spun off from the Wrangler research line [18]. Table 3 shows the quality of estimates for our approach and the baselines. We measure the quality of the predictions by the same criteria as the previous tasks. By these metrics, our approach improves the quality of predictions and beats the next-best approach by a clear margin: F_1 score is improved by 0.072, precision by 8.4 percentage points and recall by 6.0 percentage points.

4.4 Miscellaneous

Dataset contamination Here we perform an experiment to validate whether any of the testing data occurred in the training corpus of the large-language model, an issue called *dataset contamination* or *data leakage*. Because these models are trained on internet data [10] and we use public benchmarks, they may have seen the test data in training. We test on seven guaranteed-unseen tables (listed in the technical report [19]) and their columns, all uploaded between April–June 2023 to the federal data repository Data.gov. They are guaranteed-unseen because the foundation model training was completed on or before March 2023. Repeating the supervised column-type annotation task as in Section 4.2, we measure a 0.857 F_1 score, 90.0% precision and 81.8% recall. This is within 0.01 F_1 points, 0.1% precision and 5% recall of the benchmark results.

Open-source models To demonstrate the versatility of this approach, we run CHORUS with three alternative, open-source foundation models on the table-class detection task. We consider Vicuna [40], a variant of LLaMA [28], and the more advanced model LLaMA 2 [29]. Table 4 shows the results. While OpenAI's GPT model performs best, the best open-source model is very competitive. LLaMA 2 outperforms the best baseline model for this task—DoDuo-Wiki—by 0.136 F_1 points, on precision by 13.6 percentage points and on recall by 9.6 percentage points. This model lags behind the GPT model by only 0.03 F_1 points.

[0]This work is supported by the National Science Foundation under Grants NSF-BSF 2109922 and NSF IIS 2314527.

References

[1] Nora Abdelmageed, Jiaoyan Chen, Vincenzo Cutrona, Vasilis Efthymiou, Oktie Hassanzadeh, Madelon Hulsebos, Ernesto Jiménez-Ruiz, Juan Sequeda, and Kavitha Srinivas. Results of semtab 2022. In Vasilis Efthymiou, Ernesto Jiménez-Ruiz, Jiaoyan Chen, Vincenzo Cutrona, Oktie Hassanzadeh, Juan Sequeda, Kavitha Srinivas, Nora Abdelmageed, and Madelon Hulsebos, editors, *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, SemTab 2021, co-located with the 21st International Semantic Web Conference, ISWC 2022, Virtual conference, October 23-27, 2022*, volume 3320 of *CEUR Workshop Proceedings*, pages 1–13. CEUR-WS.org, 2022. URL `https://ceur-ws.org/Vol-3320/paper0.pdf`.

[2] Inc. Anaconda. State of data science. https://www.anaconda.com/resources/whitepapers/state-of-data-science-2021, July 2021.

[3] Jacob Andreas. Language models as agent models, 2022.

[4] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language models enable simple systems for generating structured views of heterogeneous data lakes, 2023.

[5] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL `https://arxiv.org/abs/2108.07258`.

[6] Zhimin Chen, Vivek R. Narasayya, and Surajit Chaudhuri. Fast foreign-key detection in microsoft SQL server powerpivot for excel. *Proc. VLDB Endow.*, 7(13):1417–1428, 2014. doi: 10.14778/2733004.2733014. URL `http://www.vldb.org/pvldb/vol7/p1417-chen.pdf`.

[7] Tamraparni Dasu, Theodore Johnson, S. Muthukrishnan, and Vladislav Shkapenyuk. Mining database structure; or, how to build a data quality browser. In Michael J. Franklin, Bongki Moon, and Anastassia Ailamaki, editors, *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA, June 3-6, 2002*, pages 240–251. ACM, 2002. doi: 10.1145/564691.564719. URL `https://doi.org/10.1145/564691.564719`.

[8] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. TURL: table understanding through representation learning. *Proc. VLDB Endow.*, 14(3):307–319, 2020. doi: 10.5555/3430915.3442430. URL `http://www.vldb.org/pvldb/vol14/p307-deng.pdf`.

[9] Jesse Dodge, Maarten Sap, Ana Marasovic, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1286–1305. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.98. URL `https://doi.org/10.18653/v1/2021.emnlp-main.98`.

[10] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027, 2021. URL `https://arxiv.org/abs/2101.00027`.

[11] Mike Gualtieri and Noel Yuhanna. *The Forrester Wave: Big Data Hadoop Distributions, Q1 2016*. Forrester Research, Inc., January 2016.

[12] P. Bryan Heidorn. Shedding light on the dark data in the long tail of science. *Library trends*, 57(2): 280–299, 2008.

[13] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=rygGQyrFvH`.

[14] Kevin Hu, Neil Gaikwad, Michiel Bakker, Madelon Hulsebos, Emanuel Zgraggen, César Hidalgo, Tim Kraska, Guoliang Li, Arvind Satyanarayan, and Çağatay Demiralp. Viznet: Towards a large-scale visualization learning and benchmarking repository. In *Proceedings of the 2019 Conference on Human Factors in Computing Systems (CHI)*. ACM, 2019.

[15] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022.

[16] Madelon Hulsebos, Kevin Zeng Hu, Michiel A. Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César A. Hidalgo. Sherlock: A deep learning approach to semantic data type detection. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1500–1508. ACM, 2019. doi: 10.1145/3292500.3330993. URL `https://doi.org/10.1145/3292500.3330993`.

[17] Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. TABBIE: pretrained representations of tabular data. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3446–3456. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.270. URL `https://doi.org/10.18653/v1/2021.naacl-main.270`.

[18] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. Wrangler: interactive visual specification of data transformation scripts. In Desney S. Tan, Saleema Amershi, Bo Begole, Wendy A. Kellogg, and Manas Tungare, editors, *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 3363–3372. ACM, 2011. doi: 10.1145/1978942.1979444. URL `https://doi.org/10.1145/1978942.1979444`.

[19] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. CHORUS: foundation models for unified data discovery and exploration. *CoRR*, abs/2306.09610, 2023. doi: 10.48550/arXiv.2306.09610. URL `https://doi.org/10.48550/arXiv.2306.09610`.

[20] Aneta Koleva, Martin Ringsquandl, Mitchell Joblin, and Volker Tresp. Generating table vector representations. *CoRR*, abs/2110.15132, 2021. URL `https://arxiv.org/abs/2110.15132`.

[21] Vladimir Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966.

[22] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories, 2022.

[23] Avanika Narayan, Ines Chami, Laurel J. Orr, and Christopher Ré. Can foundation models wrangle your data? *Proc. VLDB Endow.*, 16(4):738–746, 2022. URL `https://www.vldb.org/pvldb/vol16/p738-narayan.pdf`.

[24] Fatemeh Nargesian, Ken Q. Pu, Bahar Ghadiri Bashardoost, Erkang Zhu, and Renée J. Miller. Data lake organization. *IEEE Trans. Knowl. Data Eng.*, 35(1):237–250, 2023. doi: 10.1109/TKDE.2021.3091101. URL `https://doi.org/10.1109/TKDE.2021.3091101`.

[25] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022. doi: 10.48550/arXiv.2203.02155. URL `https://doi.org/10.48550/arXiv.2203.02155`.

[26] Dominique Ritze and Christian Bizer. Matching web tables to dbpedia - A feature utility study. In Volker Markl, Salvatore Orlando, Bernhard Mitschang, Periklis Andritsos, Kai-Uwe Sattler, and Sebastian Breß, editors, *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017*, pages 210–221. OpenProceedings.org, 2017. doi: 10.5441/002/edbt.2017.20. URL `https://doi.org/10.5441/002/edbt.2017.20`.

[27] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. Annotating columns with pre-trained language models. In *Proceedings of the 2022 International Conference on Management of Data*. Association for Computing Machinery, 2022. ISBN 9781450392495. URL `https://doi.org/10.1145/3514221.3517906`.

[28] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/arXiv.2302.13971. URL `https://doi.org/10.48550/arXiv.2302.13971`.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/arXiv.2307.09288. URL `https://doi.org/10.48550/arXiv.2307.09288`.

[30] Trifacta. Trifacta wrangler. `https://cloud.trifacta.com`, 2023. Accessed: 2023-04-10.

[31] Immanuel Trummer. Can deep neural networks predict data correlations from column names? *CoRR*, abs/2107.04553, 2021. URL `https://arxiv.org/abs/2107.04553`.

[32] Immanuel Trummer. Towards nlp-enhanced data profiling tools. In *12th Conference on Innovative Data Systems Research, CIDR 2022, Chaminade, CA, USA, January 9-12, 2022*. www.cidrdb.org, 2022. URL `https://www.cidrdb.org/cidr2022/papers/a55-trummer.pdf`.

[33] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. Technical Report MSR-TR-2023-8, Microsoft, February 2023. URL `https://www.microsoft.com/en-us/research/publication/chatgpt-for-robotics-design-principles-and-model-abilities/`.

[34] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

[35] Cong Yan and Yeye He. Auto-suggest: Learning-to-recommend data preparation steps using data science notebooks. In David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo, editors, *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1539–1554. ACM, 2020. doi: 10.1145/3318464.3389738. URL `https://doi.org/10.1145/3318464.3389738`.

[36] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8413–8426. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.745. URL `https://doi.org/10.18653/v1/2020.acl-main.745`.

[37] Ce Zhang, Jaeho Shin, Christopher Ré, Michael J. Cafarella, and Feng Niu. Extracting databases from dark data with deepdive. In Fatma Özcan, Georgia Koutrika, and Sam Madden, editors, *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 847–859. ACM, 2016. doi: 10.1145/2882903.2904442. URL `https://doi.org/10.1145/2882903.2904442`.

[38] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çagatay Demiralp, and Wang-Chiew Tan. Sato: Contextual semantic type detection in tables. *Proc. VLDB Endow.*, 13(11):1835–1848, 2020. URL `http://www.vldb.org/pvldb/vol13/p1835-zhang.pdf`.

[39] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball, 2023.

[40] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685, 2023. doi: 10.48550/arXiv.2306.05685. URL `https://doi.org/10.48550/arXiv.2306.05685`.