

Databases: From Data Storage Towards Partners for Information Access and Discovery

Benjamin Hättasch^{1,2}, Carsten Binnig^{2,1}

¹German Research Center for Artificial Intelligence (DFKI), Germany

²Technical University of Darmstadt (TU Darmstadt), Germany

Abstract

Classically, data storage and its usage were separated: There were databases with the task of storing large amounts of data efficiently and allowing fast access. On the other hand, there were experts who built tools to access or manipulate certain parts of that data to solve very specific tasks. With the ever-growing amount of data, users need new semantic sense-making methods that reduce that overhead. Automation may help here—and indeed, in the last years, we have witnessed a strong growth of easily usable approaches for information access, especially through the wide adaptation of ChatGPT in society. However, these approaches often rely on diffuse background knowledge and merely use the available structured data. As a result, they take a lot of control from the user and, nevertheless, cannot provide quality guarantees or traceable results. In this position paper, we therefore argue that simple automation, through LLMs or other means, is insufficient. Instead, we propose building systems that leverage structured data and user interaction, and automate some tasks while still leaving the users in control through carefully designed means of interaction. Based on three case studies, we analyze how treating the data system as a partner could not only improve performance, but make relevant information more accessible for all kinds of users, too. In this regard, we identify directions and principles for future research.

Keywords

databases, data systems, information access, human automation interaction, human data interaction, interaction with automation, human AI interaction, human-centered AI (HCAI)

1. Introduction

The need to democratize data systems Accessing, processing, and storing relevant information is important in many fields, from research, healthcare, and public service to journalism: Researchers need to find relevant existing work. City administrations want to learn about the needs of their citizens from large numbers of complaints. Fiscal authorities try to uncover tax evasion and money laundering. Journalists need facts and statistics to back their articles. Healthcare professionals interpret test results and patient files to learn about their patients and how to help them.

Meeting these needs at scale requires the automation of knowledge tasks. Computers, with their ability to quickly store and process enormous amounts of data and the progress in AI (e.g., pattern detection, automatic translation, language modeling) offer great new opportunities for that [1]. Unfortunately, those approaches are often associated with high effort and overhead, and can only be used by AI experts. A focus must, therefore, be set on not just providing the general functionality but making it accessible to a wide range of people. The importance is underlined by the growing demand for data scientists that currently cannot be met by far [2, 3, 4], slowing down scientific, industrial, and societal development and progress.

The need for carefully designed automation As a consequence, we witnessed a rise in approaches that are very easy to use: End users experience tools to directly receive answers to their (knowledge) questions (e.g., Siri, Google Assistant, ChatGPT) in their day-to-day life. It, therefore, seems only natural that they expect something similar for their professional tasks. Hence, many tools with conversational interfaces have emerged recently, e.g., to query data sources in natural language or create code blocks

AutomationXP25: Hybrid Automation Experiences, April 27, 2025, Yokohama, Japan. In conjunction with ACM CHI'25

✉ benjamin.haettasch@dfki.de (B. Hättasch); carsten.binnig@cs.tu-darmstadt.de (C. Binnig)

id 0000-0001-8949-3611 (B. Hättasch); 0000-0002-2744-7836 (C. Binnig)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

to access, process, and visualize information. However, from our perspective, these approaches often overshoot the target by taking away that much control from the users, leading to (unjustified) blind trust in the results [5] and a lack of efficient means to correct errors in the results or resulting behavior.

While some automation is needed to solve the relevant tasks at scale, full automation is often impractical. After all, this is not autonomous driving. Those interacting with the data should still be in the “driver’s seat”—and hence need to be provided with useful interfaces to interact and control the process while being relieved of tedious tasks. Thus, it is important to carefully design the interaction and balance between control and automation to prevent, as described by Bainbridge [6], the irony that automation might make a user’s tasks more difficult instead of supporting them. Wiberg and Bergqvist [7] describe how automation requires shifting from direct control towards cooperation with the computer. This can help ensure that humans identify with their work and feel responsible for the results [8]. Designing this interaction to make it feel like working with a team partner seems promising, but achieving it is not easy.

The need for integrated solutions These improved data systems should focus on reducing manual work, not necessarily the amount of interaction. Combining research on AI and UX to design systems where AI complements interaction instead of replacing it can be a key to making systems more accessible and useful [7]. We argue that including the data by integrating the semantic functionality directly into the data system can improve usability and quality even further. This holds, e.g., for conversational agents, where the number of necessary interactions can be reduced when the computer has access to data characteristics of the underlying data storage system [9]. Building systems that adapt to domains and users based on interaction may also be a good alternative to resolve drawbacks of using LLMs, such as high computation efforts, legal or privacy problems, lack of guarantees, or missing resources for fine-tuning. Again, this requires a close integration with relevant data sources.

Since the quality of this integration will influence the quality of the results and the answering speed of the system, it will change how the user perceives the computer it is currently collaborating with (is it more of an unskilled assistant or an expert?), affecting trust and adaptation. Furthermore, we argue that interaction in integrated systems can be shaped to achieve better quality compared to other automated approaches with the same amount of manual work for the users. Thus, this might help to overcome zero-sum assumptions, as criticized by Shneiderman [10].

Finally, it has long been recognized that users need to be involved in processes to avoid boredom and distraction and thus poorer results overall [11]. Designing integrated data systems instead of leaving it up to the user to combine multiple tools can help to achieve such a balanced interaction.

2. Case Studies

In this section, we will demonstrate the advantages of integrated interactive data systems mentioned above. We will collect requirements for these systems and sketch potential interaction patterns. To do so, we present three case studies of tasks from the data systems community and discuss how interactive systems could tackle them.

First, we present an interactive system for ad-hoc information extraction that we built in the last few years. This system is targeted at people without a background in data or computer science. Through interaction, this system can outperform learned and few-shot approaches and that at only a fraction of computing costs. Afterwards, we will describe a tool to increase the productivity of experts dealing with large data systems and analyze why existing non-interactive approaches cannot solve that problem. Finally, we present a vision of how data storage and access could radically change and how experts and non-experts will benefit.

2.1. Table Extraction from Text

Why? Large amounts of information are only available as written text. Users like journalists or researchers who are confronted with a text collection are often only interested in specific facts from

it, e.g., they need a table of persons mentioned together with their birthplaces or of the prevailing weather conditions during described events. Such a structured representation allows them to leverage the contained knowledge without having to read those texts repeatedly. However, many data discovery techniques require technical background knowledge and are not easily accessible; there is a lack of easy-to-use tools for information extraction, as information overload can limit the performance of knowledge workers and not all existing tools provide sufficient benefit [12].

Information needs differ across users and situations, ranging from facts that are short and often expressed in similar terms (like the date of an event) to long phrases (like weather conditions), which can be described in various ways. Therefore, building a one-size-fits-all system that covers all possible information needs and domains is very difficult—custom extraction pipelines are necessary. Unfortunately, crafting domain-specific rules, training custom extractors, or prompting an LLM for each source document is costly and requires time, expertise, and suitable data.

How? Therefore, we built a system that allows users to extract tables interactively and even run *SQL* queries over text collections in an ad-hoc manner [13]. Its core feature is the interaction between the user and the computer, allowing the computer to adapt to the domain without needing explicit instructions. The computer will create an initial version of the requested table and then present an excerpt to request feedback from the user. The user can then confirm cells of that table or point the computer towards the right solution by selecting the correct span from one of the source texts to fill that cell. The computer will then update other parts of the table based on that feedback and present another intermediate result for feedback. By carefully selecting which rows to present for feedback, the computer can steer for what it receives feedback. On the other hand, we use human abilities to quickly identify standout values by leaving them the choice of what exactly to feedback from a subset. This process repeats until the user is satisfied with the extraction quality (the requirements for this can vary greatly depending on the scenario). Our approach allows non-expert users to automatically extract and organize relevant content from large text collections using a simple graphical interface without the need for programming skills.

What are the advantages? Our interactive approach allows quickly adapting to a new domain without requiring extensive and costly training. The users do not have to explain what the names of potentially very domain-dependent column titles mean, nor do they have to provide additional resources like manually annotated training data. Our system supports aggregation operations like counting and summing up. It thus can directly produce tables stating information that is not explicitly mentioned in the documents and hence not discoverable by pure extraction or search approaches. Our experiments show that the number of required interactions depends on a well-suited strategy for selecting rows for feedback.¹ Throughout this interaction, the resulting table-filling quality will be much better than for a non-interactive (e.g., few-shot) system where the user provides the same amount of annotations in an unguided process [13]. By using user-provided samples for more relevant language understanding tasks, that matching quality can be increased even further [15]. At the same time, the required computation will be magnitudes lower than for running the complete large text collections through large generative models. Thus, in summary, an integrated approach with a simple user interface makes the functionality available to more people, can reduce costs for usage, and might lead to better quality of the results.

2.2. “Sloppy SQL”: Supporting near-correct queries

Why? Accessing or manipulating data in a database traditionally requires writing *SQL* queries. In the last decade, *natural language interfaces for databases (NLDBs)* emerged as an alternative, treating the problem either as a single natural language to *SQL* translation task [16, 17, 18] or building conversational interfaces that allow multi-turn interaction [19, 9]. However, even with the usage of large language

¹A similar effect can be observed in the training method of LLM distilling [14], where a small model selects examples for labeling, and the resulting model then outperforms a model trained on a superset.

models like *ChatGPT*, the translation quality stays substantially below the one of human experts [20, 21, 22]. Hence, while these *NLDBs* are very useful to make data accessible for non-experts, there are scenarios where the quality reached by automatic translation is not sufficient. It seems more promising to concentrate on making it easier for experts to craft *SQL* queries.

Real-world database schemas can consist of thousands of wide tables with hundreds of columns, rendering it nearly impossible to keep the details in mind. Moreover, queries are often crafted by consultants who might have a general model of the domain they are working on in mind but do not know the specific design of the organization for which they are creating the reports. An interactive tool that allows them to enter a *sloppy* version of the query and then develop the correct one together with the computer would be very helpful. As *sloppy* we define syntactically correct queries that use, e.g., synonyms of correct table or column names, are incomplete, or even assume a differing schema.

How? The interaction could look as follows: The user first inserts a version of the query based on their mental domain knowledge. The computer then tries to map between the user input and the database schema, replacing table and column names. At the same time, it selects additional information for the user, such as displaying possibly relevant table snippets, characteristics and visualizations of the table contents, or results and error messages from executing the query or parts of it. The user can then review the proposed changes, choose between options, or manually refine parts of the query. This process repeats until the user is convinced that their query is correct and works as intended.

What is currently missing? One could argue that there is no need for an integrated, interactive system here, and that the problem could instead be seen as a simple translation from sloppy to correct queries. We therefore tested how well different versions of OpenAI’s GPT as well as a Llama 3 model [23] fine-tuned for this task perform when provided with an erroneous query and the real database schema and are instructed to fix the query. For that, we used modified (*sloppy*) versions of queries from the Spider [24] and BIRD [22] benchmarks, and additionally manually crafted a subset of 100 queries from them for which we ensured that all information necessary to correct the query is present to the system. We tried different prompts and different formats for representing the database schema, included chain-of-thought, and even built a multi-turn version where the corrected query, together with any error messages from execution, are presented to the LLM again. Nevertheless, throughout all our experiments, the execution accuracy stayed below 40 % on all models.

How could it be instead? Therefore, we propose to tackle the problem using an interactive system, which offers the following opportunities: The repeated interaction between the user and the computer should allow it to converge to a good solution for a problem that cannot (yet) be solved by the computer alone. From the perspective of a programmer, this interaction pattern much more resembles pair-programming and, therefore, a process where one works together in a team instead of the instructive style of giving orders and hoping they lead to the correct result when prompting tools like GitHub Copilot. Furthermore, such a system could directly incorporate specific database schemas and even contents to (faster) produce better results. Finally, an interactive system crafted directly for this task could display headers and parts of the contents, or results for (parts of) the queries (choosing automatically which of them are relevant based on the interaction with the system). Working with a computer as a team partner will likely lead to better results and increase the satisfaction of the person working on the problem.

2.3. Self-Organizing Databases

Why? Governments, small organizations, and individuals often need to handle, store, and analyze incoming data but lack the knowledge and resources to design suitable storage systems and pipelines for analysis. For example, the government of a small city might ask its citizens to report any problems they notice, such as broken streetlights, trash in the park, or unfavorable traffic conditions. These reports can happen in various formats, some containing a few lines of text or consisting solely of

an image, some with more details in an already semi-structured format. A form with required fields can help enforce a consistent level of quality and completeness but might drastically reduce users' willingness to report compared to a completely unstructured transmission method, such as a WhatsApp number to message. Besides, it might not be obvious upfront which meta information needs to be requested from the user. As a result, this incoming data will most likely be stored in a way that prevents direct and convenient analysis and actions. Moreover, the city might already have existing structured information, e.g., maintenance logs or the schedule of garbage collections, but it is difficult to link it with the incoming unstructured data.

How? We, therefore, envision self-organizing data systems. Automatic systems that develop suitable schemas and even adapt the contents themselves (e.g., to normalize them) could support humans responsible for making sense of or ensuring the correct storage of data. They would not only reduce manual workload but also allow for continuous design adjustment and improvement.

What is needed? At the core, such a self-organizing database thus requires a component that continuously re-evaluates its state based on recent and all previous inputs and queries. This requires solving a wide range of challenges, which comprise i.a.: (1) Finding the right place to insert data, which might already require schema updates for new types of data or information pieces that were not available for previous rows of that semantic type. (2) Deriving structured representations from the inputs and unifying them across modalities. (3) Inferring relevant columns to answer a query and deciding whether they should be materialized or computed ad-hoc for answering that query only.

Why interaction plays a crucial role for that There are two central sources from which to draw the requested information: The data itself on the one hand and the queries of the users working with the data on the other. However, only concentrating on the data itself seems not sufficient: existing approaches tend to store all incoming data disorganized in data lakes and then focus on retrieving or producing matching tables for a specific information need posed by the users [25, 26, 27, 28]. In contrast, we aim to directly transform all incoming data into a semantically meaningful representation, starting with a very rough version that only reflects the basic concepts of the relevant domain and then refining it based on the needs of those interacting with it. Thus, although we need to preserve raw data for later database adjustments, the main surface of interaction should be the database automatically created from the input data. Adapting the data system requires considering the interaction with the system, adjusting to domain-specific wording, learning which parts of the information have to be combined, and being able to inform the people administrating the system what information might be missing.

Users can manually explore databases resulting from such a system without a clear information need in mind, as they might do with a database designed by a human data engineer. At the same time, data engineers are supported in their work so they can concentrate on semantically relevant tasks like adding new data sources instead of routine work like adapting the schema and rewriting queries.

3. What should the interaction with future data systems look like?

Findings from the case studies To summarize the main findings from the case studies, interaction can be used to provide domain adaptation or to allow computers to choose what is relevant automatically. By this, users can be supported, and overall, better results can be achieved. Furthermore, this might reduce the knowledge and skills needed to use such a system, making it accessible to further (groups of) people and allowing its application in further scenarios. By reducing manual workload, users can concentrate on semantically relevant tasks instead of routine work, resulting in better scaling and faster/more frequent results and updates. The interaction pattern needs to balance control and automation to both gain good results and increase user satisfaction.

This interaction can be shaped in many different ways: Users can enter definitions or prompts. Alternatively, systems can automatically adapt to the user based on provided examples, feedback, or

their behavior. Furthermore, it will often be relevant to choose what information to present to the user automatically. This may include explanations, justifications, confidences, visualizations, error messages, and any additional information relevant for concrete information needs or interesting for exploratory scenarios.

Since the scenarios are manifold, there is no one-size-fits-all solution for interaction patterns or interfaces for these problems; they must always be adapted to the task. However, our case studies hint that iterative approaches might often be helpful, and that it is often better when the user feels like working in a team with the computer instead of giving it orders.

Together is better Interactive data systems can, in particular, directly incorporate existing (structured) data sources for grounded results and optimized paths to them (e.g., by exploiting data characteristics to narrow down ambiguities efficiently). Integrating these data sources and automated approaches that react to the user's behavior can reduce the overall amount of explicit prompts, annotation, and additional data needed. This might allow all kinds of users to explore data freely (even without a specific information need) or even make it possible to process substantial amounts of data in a specific domain in the first place. Thus, we advocate not tackling interaction, semantic interpretation, and data management separately but considering them together when building new semantic data systems. This, however, leads to new challenges:

How should interfaces look like? Often, interface design does not play a central role in data systems research. As part of this integration, this has to evolve. It will not be sufficient to simply choose between a GUI and a conversational interface; researchers must also carefully consider which information to select and how to display it. They have to ensure that the users have the relevant information but do not get overwhelmed—which includes adapting to potentially different target groups. The resulting interfaces should relieve users from tedious tasks but simultaneously prevent blind trust and challenge users to reflect the computer's suggestions [5]. They must offer transparency and explainability, but prevent information overload, which might lead to “pseudo-control” and “pseudo-accountability” [8]. Here, a close collaboration with the HCI community and their decades of experience with designing interaction will be key.

How can this be evaluated? The second, probably most prominent question will be how to evaluate these systems. The data systems community strongly focuses on measuring the accuracy of results and the performance of their creation; this must be combined with measuring usability and user satisfaction. Joint measures may then help assess how well a system is suited for real-world applications much better than inspecting results from performance and usability evaluations individually. However, it will be necessary to develop techniques (e.g., through simulation and standardized tasks and measures) for conducting these evaluations without massively increasing the necessary efforts of the involved researchers.

We are confident that combining approaches from these different areas of research can lead to exciting new results. Therefore, we consider it very important to develop answers to the points above in order to make progress here.

Acknowledgments

Partially funded by the German Federal Ministry of Education and Research within the “The Future of Value Creation – Research on Production, Services and Work” program (grant 02L19C150) and by the Hessian Ministry of Higher Education, Research, Science and the Arts.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check, improve writing style. Further, the authors used DeepL in order to: Text Translation. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] C. Coombs, D. Hislop, S. K. Taneva, S. Barnard, The strategic impacts of Intelligent Automation for knowledge and service work: An interdisciplinary review, *The Journal of Strategic Information Systems* 29 (2020) 101600. doi:10.1016/j.jsis.2020.101600.
- [2] T. H. Davenport, D. J. Patil, Is Data Scientist Still the Sexiest Job of the 21st Century?, *Harvard Business Review* (2022). URL: <https://hbr.org/2022/07/is-data-scientist-still-the-sexiest-job-of-the-21st-century>.
- [3] Bureau of Labor Statistics, U.S. Department of Labor, Data Scientists : Occupational Outlook Handbook : U.S. Bureau of Labor Statistics, 2023. URL: <https://www.bls.gov/ooh/math/data-scientists.htm>.
- [4] World Data Science Initiative, Why Data Science is the most in-demand skill now and how can you prepare for it?, 2023. URL: <https://www.worlddatascience.org/blogs/why-data-science-is-the-most-indemand-skill-now-and-how-can-you-prepare-for-it>.
- [5] S. Müller, M. Baldauf, P. Fröhlich, AI-Assisted Document Tagging - Exploring Adaptation Effects among Domain Experts, in: P. Fröhlich, M. Baldauf, P. Palanque, V. Roto, F. Paternó, W. Ju, M. Tscheligi (Eds.), *Proceedings of the Workshop on Intervening, Teaming, Delegating*, volume 3394 of *CEUR Workshop Proceedings*, CEUR, Hamburg, Germany, 2023.
- [6] L. Bainbridge, Ironies of automation, *Automatica* 19 (1983) 775–779. doi:10.1016/0005-1098(83)90046-8.
- [7] M. Wiberg, E. S. Bergqvist, User Experience (UX) meets Artificial Intelligence (AI) - Designing Engaging User Experiences Through 'Automation of Interaction', *AutomationXP22: Engaging with Automation*, CHI'22 (2022).
- [8] S. Sadeghian, M. Hassenzahl, On Autonomy and Meaning in Human-Automation Interaction, *AutomationXP23: Intervening, Teaming, Delegating - Creating Engaging Automation Experiences*, CHI '23, April 23rd, Hamburg, Germany (2023).
- [9] M. Gassen, B. Hättasch, B. Hilprecht, N. Geisler, A. Fraser, C. Binnig, Demonstrating CAT: synthesizing data-aware conversational agents for transactional databases, *Proc. of the VLDB Endow.* 15 (2022) 3586–3589. URL: <https://www.vldb.org/pvldb/vol15/p3586-h%e4ttasch.pdf>.
- [10] B. Shneiderman, *Human-Centered AI*, Oxford University Press, Oxford, New York, 2022.
- [11] W. C. Harris, P. A. Hancock, E. J. Arthur, J. K. Caird, Performance, workload, and fatigue changes associated with automation, *The International Journal of Aviation Psychology* 5 (1995) 169–185. doi:10.1207/s15327108ijap0502_3.
- [12] P. Karr-Wisniewski, Y. Lu, When more is too much: Operationalizing technology overload and exploring its impact on knowledge worker productivity, *Computers in Human Behavior* 26 (2010) 1061–1072. URL: <https://www.sciencedirect.com/science/article/pii/S0747563210000488>. doi:10.1016/j.chb.2010.03.008, advancing Educational Research on Computer-supported Collaborative Learning (CSCL) through the use of gStudy CSCL Tools.
- [13] B. Hättasch, J. Bodensohn, L. Vogel, M. Urban, C. Binnig, Wannadb: Ad-hoc SQL queries over text collections, in: B. König-Ries, S. Scherzinger, W. Lehner, G. Vossen (Eds.), *Datenbanksysteme für Business, Technologie und Web (BTW 2023)*, 20. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme" (DBIS), 06.-10, März 2023, Dresden, Germany, *Proceedings*, volume P-331 of *LNI, Gesellschaft für Informatik e.V.*, 2023, pp. 157–181. doi:10.18420/BTW2023-08.
- [14] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, T. Pfister,

- Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes, 2023. doi:10.48550/arXiv.2305.02301. arXiv:2305.02301.
- [15] B. Hättasch, C. Binnig, More of that, please: Domain adaptation of information extraction through examples & feedback, in: HILDA'24 Workshop at SIGMOD'24, 2024, pp. 1–7. doi:10.1145/3665939.3665966.
 - [16] H. Kim, B.-H. So, W.-S. Han, H. Lee, Natural language to SQL: Where are we today?, *Proceedings of the VLDB Endowment* 13 (2020) 1737–1750. doi:10.14778/3401960.3401970.
 - [17] B. Qin, B. Hui, L. Wang, M. Yang, J. Li, B. Li, R. Geng, R. Cao, J. Sun, L. Si, F. Huang, Y. Li, A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions, 2023. doi:10.48550/arXiv.2208.13629. arXiv:2208.13629.
 - [18] G. Katsogiannis-Meimarakis, G. Koutrika, A survey on deep learning approaches for text-to-SQL, *The VLDB Journal* 32 (2023) 905–936. doi:10.1007/s00778-022-00776-8.
 - [19] T. Yu, R. Zhang, M. Yasunaga, Y. C. Tan, X. V. Lin, S. Li, I. L. Heyang Er, B. Pang, T. Chen, E. Ji, S. Dixit, D. Proctor, S. Shim, V. Z. Jonathan Kraft, C. Xiong, R. Socher, D. Radev, SPaRC: Cross-domain semantic parsing in context, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019, pp. 4511–4523.
 - [20] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, J. Zhou, Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation, *arXiv* (2023). doi:10.48550/arXiv.2308.15363. arXiv:2308.15363.
 - [21] S. Sun, Y. Zhang, J. Yan, Y. Gao, D. Ong, B. Chen, J. Su, Battle of the Large Language Models: Dolly vs LLaMA vs Vicuna vs Guanaco vs Bard vs ChatGPT – A Text-to-SQL Parsing Comparison, 2023. doi:10.48550/arXiv.2310.10190. arXiv:2310.10190.
 - [22] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, B. Wang, B. Qin, R. Geng, N. Huo, et al., Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls, *Advances in Neural Information Processing Systems* 36 (2024).
 - [23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: Open and Efficient Foundation Language Models, 2023. doi:10.48550/arXiv.2302.13971. arXiv:2302.13971.
 - [24] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, D. Radev, Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018, pp. 3911–3921.
 - [25] S. Zhang, K. Balog, Ad hoc table retrieval using semantic similarity, in: *Proceedings of the 2018 World Wide Web Conference, WWW '18*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, p. 1553–1562. URL: <https://doi.org/10.1145/3178876.3186067>. doi:10.1145/3178876.3186067.
 - [26] M. Trabelsi, Z. Chen, S. Zhang, B. D. Davison, J. Heflin, Strubert: Structure-aware bert for table search and matching, in: *Proceedings of the ACM Web Conference 2022, WWW '22*, Association for Computing Machinery, New York, NY, USA, 2022, p. 442–451. URL: <https://doi.org/10.1145/3485447.3511972>. doi:10.1145/3485447.3511972.
 - [27] F. Wang, K. Sun, M. Chen, J. Pujara, P. Szekely, Retrieving complex tables with multi-granular graph representation learning, in: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21*, Association for Computing Machinery, New York, NY, USA, 2021, p. 1472–1482. URL: <https://doi.org/10.1145/3404835.3462909>. doi:10.1145/3404835.3462909.
 - [28] J. Bodensohn, C. Binnig, Rethinking table retrieval from data lakes, in: *Proceedings of the Seventh International Workshop on Exploiting Artificial Intelligence Techniques for Data Management, aiDM 2024*, Santiago, Chile, 14 June 2024, ACM, 2024, pp. 2:1–2:5. URL: <https://doi.org/10.1145/3663742.3663972>. doi:10.1145/3663742.3663972.