Text-to-SQL with Large Language Models: Exploring the Promise and Pitfalls

Luca Sala[1,*,†], Giovanni Sullutrone[1,†] and Sonia Bergamaschi[1,†]

[1]*University of Modena and Reggio Emilia, UNIMORE*

Abstract

The emergence of Large Language Models (LLMs) represents a fundamental change in the ever-evolving field of natural language processing (NLP). Over the past few years, the enhanced capabilities of these models have led to their widespread use across various fields, in both practical applications and research contexts. In particular, as data science intersects with LLMs, new research opportunities and insights emerge, notably in translating text into Structured Query Language (Text-to-SQL). The application of this technology to such task poses a unique set of opportunities and related issues that have significant implications for information retrieval. This discussion paper delves into these intricacies and limitations, focusing on challenges that jeopardise efficacy and reliability. This research investigates the scalability, accuracy, and concerning issue of hallucinated responses, questioning the trustworthiness of LLMs. Furthermore, we point out the limits of the current usage of test dataset created for research purposes in capturing real-world complexities. Finally, we consider the performance of Text-to-SQL with LLMs from different perspectives. Our investigation identifies the key challenges faced by LLMs and proposes viable solutions to facilitate the exploitation of these models to advance data retrieval, bridging the gap between academic researcher and real-world application scenarios.

1. Introduction

In recent years, natural language processing (NLP) has been fundamentally changed by the rise of Large Language Models (LLMs). Models like BERT (Bidirectional Encoder Representations from Transformers) [1] and GPT (Generative Pretrained Transformer) [2], trained on massive corpora of written data, have shown impressive capabilities in grasping semantic relationships and solving complex tasks.

This has made them powerful tools for human-computer interaction that need an extensive semantic and domain knowledge in order to be up-to-par with the requirements of real-world applications. In particular, their ability to interpret natural language requests and translate them into executable SQL statements has the potential to revolutionize database querying, from bridging the gap between complex database systems and end-users to making data-driven insights more accessible to a broader audience.

Furthermore, the impressive adaptability and learning capabilities of LLMs offer promises of continuous improvement in query understanding and processing. As these models are exposed to more domain-specific data, their effectiveness in handling queries in various specialized fields, from healthcare to financial services, is expected to improve [3]. This not only enhances the accuracy of query conversion, but also opens up possibilities for personalized database interactions, where the model adjusts to the user's language and query patterns.

This paper explores the current limitations of Text-to-SQL systems powered by LLMs. It focuses on potential pitfalls and readily applicable solutions to improve performance for real-world use cases. The structure is as follows: Section 2 provides background on Text-to-SQL with LLMs; Section 3 addresses challenges, limitations, and solutions; followed by conclusions and future perspectives.

2. Background

2.1. Text-to-SQL

The inherent complexity of the Text-to-SQL task comes from the fundamental differences between natural language and SQL. Natural language is characterized by ambiguity, flexibility, and implicit context, whereas SQL adheres to a strict, formal syntax and requires explicit representation of relationships within a database schema. Early approaches relied heavily on handcrafted rules and grammars [4, 5], leading to systems that were difficult to generalize to new domains. With the rise of machine learning, new techniques started to take shape, employing elements like sequence-to-sequence models to learn the mapping between natural language and SQL, showing improved robustness compared to previous ones [6].

Since LLMs are neural network-based models pre-trained on massive text corpora, enabling them to capture rich linguistic patterns and world knowledge, their advent has further revolutionized the Text-to-SQL field. Key to their success is the Transformer architecture, which excels at processing sequential data and modeling long-range dependencies [7].

Their pre-training process exposes them to diverse language usage and domain knowledge [8] that can be readily made available to convert natural language into queries. Furthermore, LLMs can effectively model the logical structure of SQL, handling complex elements like nested structures and aggregations [9]. Notably, these models display potential for zero-shot or few-shot learning in Text-to-SQL, suggesting they can generate SQL queries for new database schemas with minimal or even no additional fine-tuning, thus increasing their adaptability [10].

The integration of LLMs with Text-to-SQL is currently a thriving area of research. Benchmarks like WikiSQL [6], Spider [11], and BIRD [9] play a crucial role in driving progress and providing standard evaluation metrics. These datasets consist of paired natural language questions and corresponding SQL translations across various domains.

Diverse strategies have been explored to harness the power of this technology. Among them [12] used an incremental pre-training procedure and fine-tuning on task-specific labeled data. Additionally, interest has been placed in In-Context learning (ICL) [2], where LLMs are prompted with natural language instructions, examples, and carefully engineered input sequences to generate the SQL output [13]. Finally, researchers are exploring hybrid approaches

that combine the strengths of LLMs with decoding constraints or intermediate representations to enhance the structure and controllability of the generated SQL queries [13, 14].

2.2. The need of Text-to-SQL

Relational databases, characterized by their efficient, structured, and reliable data management capabilities, have been instrumental in supporting transactional data storage and critical business operations for decades. In 2022, the market value of relational databases was an impressive USD 55.9 billion, with forecasts predicting a growth to USD 161.4 billion by 2032, showcasing a compound annual growth rate (CAGR) of 12.50% [15]. This substantial growth underscores the continuing reliance on relational databases in the digital age and highlights the increasing amount of data being processed and stored.

However, accessing and analyzing this vast reservoir of data poses a challenge, particularly for non-experts. The traditional method of interacting with databases through structured query languages such as SQL requires a deep understanding of database schemas and precise command syntax. Through NL querying, users can communicate with databases in plain text, bypassing the need to master complex query languages.

Integrating Text-to-SQL capabilities into data management systems can therefore significantly accelerate the data exploration process, enabling faster decision-making and insight discovery. It allows users to ask iterative questions, refine their queries based on previous results, and explore data relationships and patterns without the bottleneck of formulating precise SQL queries.

In summary, the need for Text-to-SQL technologies is driven by the growing complexity and volume of data stored in relational databases and the necessity to make this data accessible to a wider audience. As such, investing in and developing these technologies is crucial for organizations aiming to stay competitive in the data-driven landscape of the 21st century.

3. Addressing Challenges and Limitations

3.1. Response Time and Performance

In the realm of database interaction, response time, the time elapsed before receiving a query result, plays a vital role in ensuring smooth operation and a seamless user experience. The introduction of LLMs for query generation shifts our perspective on these metrics, placing emphasis on their inference[1] speed as they act as an additional translation layer between the user's requests and the extracted data. Understanding response time from this perspective requires a nuanced look at factors like Time to First Token (TTFT), which indicates the model's initial responsiveness, and Time Per Output Token (TPOT), which determines how efficiently it generates subsequent parts of the query. Together, TTFT and TPOT give us latency, a measure of the total time needed to produce a complete response or, in our case, the converted SQL query. Throughput, on the other hand, quantifies the server's ability to produce output tokens across multiple requests. While these metrics offer valuable insights, it's important to acknowledge

---

[1]Inference refers to the process of getting a response from the trained LLM model for the user's query or prompts.

that the hardware used to deploy any LLM has the biggest impact on these factors, making them highly susceptible to the specific context of application.

A significant gap in current research is the lack of direct comparisons between the time it takes a language model to create a query versus the time it takes a human to do the same task. This obscurity hinders our understanding of the potential advantages this technology offer. User expectations have been shaped by the immediate feedback search engines provide; it follows naturally that benchmarks should also account for the desire for fast responses.

Evaluating Text-to-SQL performance extends beyond the raw capabilities of the LLM; the methods used for assessment play a decisive role. Benchmarks like Spider [11] offer insufficient analysis of how models compare to human performance in this task. The BIRD benchmark [9] partially addresses this shortcoming by incorporating human ratings but omits crucial elements such as the number of attempts and time required for humans to write valid SQL queries. Incorporating these measurements would enable a more in-depth comparison between model and human efficiency.

As database complexity grows, the interplay between response time and performance becomes even more critical. Maintaining responsiveness without compromising reliability demands advanced techniques. Ironically, methods designed to improve LLM accuracy can sometimes worsen response time. The Chain of Thought (CoT) approach [16], for example, helps tackle complex queries by breaking them into sub-problems, while techniques like Least-to-Most [17] and Self-Consistency [18] involve repeated questioning to gain clarity and improve precision. Although beneficial for complex queries, this subdivision into steps introduces variability into both the computational resources needed and the overall time taken to generate a response. This presents a challenge in ensuring predictability and efficiency.

One possible workaround is to use specialized inference engine like the Language Processing Unit (LPU) introduced by Groq [19] that shows 3-18x improvements in Output Tokens Throughput compared to traditional providers. Furthermore, it guarantees consistent Time-to-First-Token reducing drastically the variability of responses.

Balancing the benefits of advanced LLM techniques with the need for predictable and efficient database interactions remains a critical area for ongoing research and development in the field of NLP and database management.

3.2. Scalability

The rapid expansion of available data and the increasing complexity of databases present significant challenges for applying LLMs to the task of Text-to-SQL. Current models struggle with large databases and real-world datasets that often contain inconsistencies or 'noisy' values [9]. Additionally, the inherent complexity of databases, combined with the limited context window which determines how much information they can hold in memory, can lead to significant compression of the prompt, hindering their understanding of the underlying data structure.

Current methodologies, in fact, base the pre-trained model's grounding on two main elements: schema linking and example value sampling.

Schema linking identifies references to database elements (tables, columns, etc.) within the natural language query to be added to the prompt [20]. As databases scale, queries may

reference a broader range of tables, making schema linking more difficult and forcing a stricter selection, impacting overall performance [9, 20].

Value sampling aims to provide the LLM with representative examples from the linked tables [10]. However, with larger tables, these samples may not adequately reflect the full distribution of data, potentially misleading the LLM.

Fortunately, the ongoing evolution of these models suggests that scalability issues may be addressed intrinsically as these models improve.

Starting from early models like GPT-3.5-turbo which had a context window of 4096 tokens and GPT-4 with 8192 tokens, significant progress has been made in GPT-3.5-turbo-16k-0613 and GPT-4-32k-0613 with their limits increased to 16384 and 32768 tokens, respectively. Two of today's most advanced models, Claude 3 [21] and Gemini 1.5 Pro [22], offer even more impressive context windows, up to 200,000 tokens for the former and up to 1 million tokens for the latter.

A potential drawback for long context models, however, is the performance drop in specific positions of their memory which could result in a loss of task-essential information. It has been observed that performance is often highest when relevant information is located at the beginning or end of the input context, while it degrades significantly otherwise [23].

However, the most recent models claim to have mitigated the problem. Gemini 1.5 Pro achieves near-perfect (>99%) recall up to multiple millions of tokens of in all modalities, i.e., text, video, and audio, and even maintaining this recall performance when extending to 10M tokens in the all three modalities [22]. Additionally Claude 3 Opus not only achieved near-perfect recall, surpassing 99% accuracy, but in some cases, it even identified the limitations of the evaluation itself by recognizing that the "needle" sentence used to test the information retrieval capability appeared to be artificially inserted into the original text by a human [21].

3.3. Hallucinations

The term "hallucinations", in the context of LLMs, refers to instances where the model generates inaccurate or misleading information. This phenomenon can arise due to various factors, such as the inherent complexities of natural language, biases within the training data, and limitations of the model itself. Hallucinations represent a challenge in the field of Text-to-SQL, where accuracy and precision in relation to the underlying database and its schema are paramount.

Within these systems, hallucinations manifest when the LLM fabricates incorrect assumptions about the database structure or invents non-existent tables, columns, or data values. These hallucinations pose a serious threat to the model's performance and reliability, as they can lead to SQL queries that are either invalid or generate incorrect results.

Researchers have observed that hallucinations involving the creation of fictional table data are a particularly prevalent issue in large-scale databases [9]. Even when schema linking techniques are employed to align the generated query with the structure of the target database, these problems persist.

Mitigating hallucinations is an active area of research that has seen various interesting proposals. Recent solutions include techniques like response selectors that use beam search [2]

---

[2]A decoding strategy that, instead of selecting only the single most likely word at each step, keeps track of multiple likely sequences

to choose executable SQL queries to use as final answer [24, 14]. Another technique is to use an output calibration step that encompasses, among others, a fuzzy search to find the closest matching columns to potentially resolve invalid ones [25]. A new avenue of research, however, is the use of Uncertainty Quantification (UQ) to assess the confidence of an LLM's generated output as UQ methods can assign confidence scores to different parts of the model's output. [26] shows empirically that UQ techniques allow relatively inexpensive fact-checking. This could have a twofold application: to highlight possible hallucinated terms in the converted query to be changed by the user or as additional information for a self-correction procedure of the model itself.

3.4. Dataset Representativity

In the realm of NLP and database query creation, various datasets and benchmarks have been developed in order to fill the gap between human language and structured database queries.

Among these, there are ATIS [27] and GEO [28] datasets which contain less than 500 unique SQL queries. On the other hand, WikiSQL [6] includes a larger number of queries and significantly larger tables, but it only covers basic queries. Spider [11] aims to address the limitations of WikiSQL by incorporating more complex, multi-table queries and a broader diversity of SQL queries, thus improving the ability of models to understand and generate intricate SQL commands from natural language inputs. Following Spider, BIRD [9] aims to further advance this domain by focusing on being more realistic in collecting data from real-world scenarios, while retaining all the complexity and variability of such data in the dataset.

However, BIRD is not without its limitations. Firstly, it exhibits bias in the generation of NL questions, primarily due to the presumed background knowledge of the user regarding the database's structure and terminology. This assumption can lead to a gap between ad-hoc and real-world query formulations, as a typical user may not recall specific details about the database or might use incorrect terms.

Including non-experts in the creation of NL questions or limiting their schema knowledge are two potential ways to mitigate these biases. This approach may guarantee a closer representation of generated queries to that of a larger user base.

Secondly, in BIRD, tables or fields not accessible due to user privileges or absence, are not explored, raising concerns about its practicality in real-world scenarios. One way to better capture real-world facets is by intentionally including non-implementable queries. This intentional introduction of real-world imperfections would enable more robust testing. To this end, we suggest introducing one promising strategy proposed in [29]. Applied in the Text-to-SQL field, this would entail fine-tuning a model on a dataset where such queries are intentionally tagged with an "I don't know" response. This approach encourages models to recognize the limits of their ability and avoid the tendency to "hallucinate" solutions that violate database constraints or permissions. The key insight is that a model capable of acknowledging its limitations is likely to be far more valuable in a practical setting than one that produces incorrect or misleading results.

Furthermore, existing Text-to-SQL datasets and benchmarks often underutilize the vast knowledge and contextual understanding capabilities of LLMs. While they excel at incorporating domain knowledge, datasets currently lack queries designed to test these abilities. Considering

that non-expect user may naturally create questions incorporating cultural references (e.g. "list movies released in the year of the dragon") or requiring the translation of colloquial terms into precise expressions (e.g. "show me sales figures for the summer months"). This gap represents a significant missed opportunity.

3.5. Knowledge Acquisition Methods

For accurate Text-to-SQL conversion in professional settings, models must incorporate field-specific linguistic, domain, and mathematical knowledge [30]. The first enables the model to deal with terminology that may be different between question and underlying schema, the second allows the conversion of domain specific concepts and the last provides the implicit mathematical or SQL operations needed to solve complex requests.

Current solutions either utilize fine-tuning (FT) or In-context Learning (ICL).

Fine-tuning is the more traditional approach for adapting LLM to specific tasks. It involves updating a pre-trained model's weights through gradient descent using a related labeled dataset. ICL, on the other hand, guides model behavior without weight updates providing input-output pairs within the prompt itself, demonstrating the desired response for the task. Both methodologies have intrinsic cost considerations.

FT in spite of the improved data efficiency thanks to the pre-trained weights, still needs a not insignificant amount of high quality labeled data to work correctly, resulting in a specialized model on the specific task at hand, hindering its use for multiple concurrent downstream tasks. Futhermore the high computational expenditure of tuning a LLM can't be ignored. This methodology, however, provides a clear view of the costs since they are limited to the additional training phase.

ICL, instead, has the drawbacks of processing the additional examples provided at each execution, increasing the memory usage and time to first token, resulting in a model which performances lag behind the fine-tuning procedure [2] and is highly sensitive to wording [31] and pair ordering [32]. The retrieval of relevant examples from a database has also to be accounted for in the resource consumption. This combination of elements makes the long-term costs and effectiveness of in-context learning more opaque.

Recently a new idea has been proposed as third option. [25] introduced the use of Parameter-Efficient Fine-Tuning (PEFT), speficially Low-Rank Adaptation (LoRa) [33], to create a model-agnostic framework to efficiently adapt pre-trained models to the task at hand by changing only a small amount of parameters. Additionally, to solve the limits of fine-tuning for multiple domains, a "Plugin hub" has been introduced to both enable the hot-swap of specialized weights to tackle different databases and plugin (i.e. weights) creation starting from merged field-related ones [25].

Regardless of the chosen methodology, a critical challenge lies in efficiently acquiring and providing the necessary field-specific knowledge to the model.

In [34, 35], different examples are annotated and used as fine-tuning source. This, however, has high generation costs since there is a need for expert human annotators to instill a diverse and accurate understanding in the model and, therefore, the data.

[36] tries to solve this by utilizing publicly available resources to retrieve relevant field information. This "bank" of knowledge is then used to guide the model towards the correct

schema linking and conversion. The proposed methodology does mitigate the incurred initial cost, but the bank creation, without constant updates or improvements, can miss useful data or lag behind fast evolving fields. Another issue is that, without careful filtering during the set up of the knowledge archive, the extraction process may generate noisy or conflicting information with a negative impact on the following retrieval operations.

One possible solution to obtain the best of both worlds would be to use the recent advancements in LLM's tool-usage to enable the creation of the bank of knowledge at run-time. In particular, we envision a pipeline where the model, given the natural language prompt, is able to actively scour the internet to extract the knowledge needed for a correct conversion. This could be both applied for augmenting existing datasets and at inference time to help translating the user's intention into query. This pipeline can also be easily merged with the proposed solution in [25] to create an ever-evolving "plugin hub" that is able to adapt to new terminologies, concepts or requirements.

4. Conclusion and Future Perspective

In this paper, we have shown that LLMs have the potential to bridge the gap between natural language and SQL queries. However, this promise demands additional research to be truly realised. While this new technology demonstrates an impressive ability to interpret and translate natural language into structured queries, it comes with several significant challenges that must be acknowledged. These include the need to effectively mitigate hallucinations, ensure scalability for complex databases, reduce response times to practical levels, and develop robust methods for integrating domain-specific knowledge. Constructing representative training datasets is also paramount, ensuring the models can adapt to diverse linguistic expressions, handle unanswerable queries, and reflect the nuances of real-world user interactions. By systematically overcoming these hurdles, we can pave the way for truly intuitive and accessible database interaction tools, fostering widespread data democratization and significantly enhancing decision-making processes across various domains.

5. Acknowledgments

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018). URL: https://arxiv.org/abs/1810.04805.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[4] C. Févotte, J. Idier, Algorithms for nonnegative matrix factorization with the β-divergence, Neural computation 23 (2011). doi:10.1162/NECO_a_00168.

[5] S. Bergamaschi, F. Guerra, S. Rota, Y. Velegrakis, A hidden markov model approach to keyword-based search over relational databases, in: Conceptual Modeling–ER 2011: 30th International Conference, ER 2011, Brussels, Belgium, October 31-November 3, 2011. Proceedings 30, Springer, 2011, pp. 411–420.

[6] V. Zhong, C. Xiong, R. Socher, Seq2sql: Generating structured queries from natural language using reinforcement learning, arXiv preprint arXiv:1709.00103 (2017).

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023. arXiv:1706.03762.

[8] A. Roberts, C. Raffel, N. Shazeer, How much knowledge can you pack into the parameters of a language model?, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 5418–5426. URL: https://aclanthology.org/2020.emnlp-main.437. doi:10.18653/v1/2020.emnlp-main.437.

[9] 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).

[10] S. Chang, E. Fosler-Lussier, How to prompt llms for text-to-sql: A study in zero-shot, single-domain, and cross-domain settings, 2023. arXiv:2305.11853.

[11] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, et al., Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task, arXiv preprint arXiv:1809.08887 (2018).

[12] H. Li, J. Zhang, H. Liu, J. Fan, X. Zhang, J. Zhu, R. Wei, H. Pan, C. Li, H. Chen, Codes: Towards building open-source language models for text-to-sql, 2024. arXiv:2402.16347.

[13] M. Pourreza, D. Rafiei, Din-sql: Decomposed in-context learning of text-to-sql with self-correction, 2023. arXiv:2304.11015.

[14] H. Li, J. Zhang, C. Li, H. Chen, Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql, 2023. arXiv:2302.05965.

[15] S. Singh, Relational database market research report: Information by type (in-memory, disk-based, and others), by deployment (cloud-based, and on-premises) by end user (bfsi, it & telecom, retail & e-commerce, manufacturing, healthcare, and others), and by region (north america, europe, asia-pacific, and rest of the world) –market forecast till 2032., https://www.marketresearchfuture.com/reports/relational-database-market-18851, 2024. Accessed: 2024-03-02.

[16] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. arXiv:2201.11903.

[17] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, E. Chi, Least-to-most prompting enables complex reasoning in large language models, 2023. arXiv:2205.10625.

[18] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou,

Self-consistency improves chain of thought reasoning in language models, 2023. arXiv:2203.11171.

[19] Inference speed is the key to unleashing ai's potential, 2024. https://wow.groq.com/inference-speed-is-the-key-to-unleashing-ai-potential/ [Accessed: (17 Mar 2024)].

[20] M. Pourreza, D. Rafiei, Din-sql: Decomposed in-context learning of text-to-sql with self-correction, Advances in Neural Information Processing Systems 36 (2024).

[21] Introducing the next generation of claude, 2024. https://www.anthropic.com/news/claude-3-family [Accessed: (13 Mar 2024)].

[22] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al., Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, arXiv preprint arXiv:2403.05530 (2024).

[23] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics 12 (2024) 157–173.

[24] A. Suhr, M.-W. Chang, P. Shaw, K. Lee, Exploring unexplored generalization challenges for cross-database semantic parsing, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8372–8388.

[25] C. Zhang, Y. Mao, Y. Fan, Y. Mi, Y. Gao, L. Chen, D. Lou, J. Lin, Finsql: Model-agnostic llms-based text-to-sql framework for financial analysis, arXiv e-prints (2024) arXiv–2401.

[26] E. Fadeeva, A. Rubashevskii, A. Shelmanov, S. Petrakov, H. Li, H. Mubarak, E. Tsymbalov, G. Kuzmin, A. Panchenko, T. Baldwin, P. Nakov, M. Panov, Fact-checking the output of large language models via token-level uncertainty quantification, 2024. arXiv:2403.04696.

[27] C. T. Hemphill, J. J. Godfrey, G. R. Doddington, The atis spoken language systems pilot corpus, in: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990, 1990.

[28] C. Finegan-Dollak, J. K. Kummerfeld, L. Zhang, K. Ramanathan, S. Sadasivam, R. Zhang, D. Radev, Improving text-to-sql evaluation methodology, arXiv preprint arXiv:1806.09029 (2018).

[29] K. Kang, E. Wallace, C. Tomlin, A. Kumar, S. Levine, Unfamiliar finetuning examples control how language models hallucinate, arXiv e-prints (2024) arXiv–2403.

[30] L. Dou, Y. Gao, X. Liu, M. Pan, D. Wang, W. Che, D. Zhan, M.-Y. Kan, J.-G. Lou, Towards knowledge-intensive text-to-sql semantic parsing with formulaic knowledge, arXiv preprint arXiv:2301.01067 (2023).

[31] A. Webson, E. Pavlick, Do prompt-based models really understand the meaning of their prompts?, 2022. arXiv:2109.01247.

[32] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, S. Singh, Calibrate before use: Improving few-shot performance of language models, 2021. arXiv:2102.09690.

[33] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. arXiv:2106.09685.

[34] Y. Wang, J. Berant, P. Liang, Building a semantic parser overnight, in: C. Zong, M. Strube (Eds.), Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 1332–1342. URL: https://aclanthology.org/P15-1129. doi:10.3115/v1/P15-1129.

[35] J. Herzig, J. Berant, Don't paraphrase, detect! rapid and effective data collection for semantic parsing, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3810–3820. URL: https://aclanthology.org/D19-1394. doi:10.18653/v1/D19-1394.

[36] L. Dou, Y. Gao, X. Liu, M. Pan, D. Wang, W. Che, D. Zhan, M.-Y. Kan, J.-G. Lou, Towards knowledge-intensive text-to-sql semantic parsing with formulaic knowledge, 2023. arXiv:2301.01067.