IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Review Paper On Text-To-SQL Generation Systems

Dr Ch Mallikarjuna Rao Department of CSE
Gokaraju Rangaraju Institute of Engineering and Technology,
Telangana, India.

Sravan Reddy
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana, India.

P.Chakradhar
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana, India.

P.Abhinay
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana, India.

S.Pavan Kumar
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana, India.

ABSTRACT

Several text-to-SQL systems have been created to bridge the gap between users and data, enabling individuals without SQL expertise to ask questions in natural language and interact with databases effectively. The progress observed in text-to-SQL tasks has contributed to advancements in deep learning methods. In order to truly advance the development of text-to-SQL systems, prior research needs to be deconstructed to comprehend the applicability and challenges of various strategies. The review paper's main goal is to give an overview of text-to-sql techniques that query data using natural language. This paper can help serve as a reference for researchers and practitioners interested in developing and applying natural language interfaces for data interaction in the era of large language models.

1. INTRODUCTION:

With the rapid growth of electronic devices, databases have emerged as the standard means for storing vast amounts of data from a variety of sources, today data resides in relational databases across various domains, including but not limited to finance, e-commerce, and healthcare . This widespread occurrence of databases emphasize the extensive usefulness of querying databases through natural language. While expert professionals may effectively access the tables using custom-written structured query languages (SQLs), a natural language (NL) interface can make it easier for a larger group of non-technical users ,individuals not familiar with SQL to access databases. Consequently, there has been a notable surge in the development of Natural Language Interfaces for Databases, aimed at translating user-provided natural language queries into SQL queries.

The process of mapping text to SQL falls under Semantic Parsing (SP) problem[4]. SP plays a crucial role in the field of natural language processing (NLP). This task involves deciphering the meaning behind natural language sentences and converting them into practical executable queries or equivalent logical forms such as SQL queries. Yet, the task of generating SQL poses a greater challenge compared to the conventional semantic parsing problem. A concise natural language query may necessitate the combination of multiple tables or the inclusion of various filtering conditions, demanding more context-based approaches.

Early efforts in this domain are based on Rule-based **Text-to-SQL** methods [1,13,14,15] employ carefully crafted templates for producing SQL queries, demonstrating effectiveness in particular scenarios. Nevertheless, these approaches heavily depend on manual rule creation, posing challenges in adapting them to diverse domains and constraining their scalability and applicability. However, with the construction of several large-scale text-to-SQL datasets, such as WikiSQL [3] and Spider [4], has paved the way for the incorporation of seq-2-seq models[8,13] and deep learning approaches into this task. This shift from rule-based methods to deep learning has been instrumental in overcoming the limitations associated with manual rule design, contributing to unprecedented performance improvements in recent years for training and developing text-to-sql systems. Growing interest have been expressed by researchers in this subject[16,17] , that focus on a systematic study of approaches implemented over the years. Recent studies[18,19] specifically focus on deep learning approaches leaving scope for latest works involving LLM based query generation techniques.

This paper has been organised as follows, section 2 provides a simple description and illustrates Text-to-SQL problem with the help of an example. Section 3 is dedicated for showcasing various kinds of datasets that have been used in the text-to-SQL scenario. Section 4 highlights various evaluation metrics and delves into the issues involved for evaluating system performance utilizing these metrics. Section 5 provides an overview of the existing research in the field. Finally, Section 6 is dedicated for conclusion.

2. PROBLEM DESCRIPTION

The Text-to-SQL (T2S) parsing problem addresses the challenge of transforming natural language queries (NLQ) into structured query language (SQL) expressions, specifically tailored for execution on a relational database (RDB). The objective is to produce a valid SQL query that captures the semantic meaning of the user's natural language question and, when executed, retrieves results aligned with the user's intent.

In Figure 1, an example of text-to-SQL conversion is depicted, showcasing the transformation of a Natural Language Query (NLQ) into a syntactically and semantically valid SQL query. This process involves consulting the pertinent schema information from the associated database, ensuring accurate alignment between the user's language input and the database structure. The system utilizes this information to generate an SQL query that precisely captures the user's intent, thereby facilitating seamless interaction with the database.

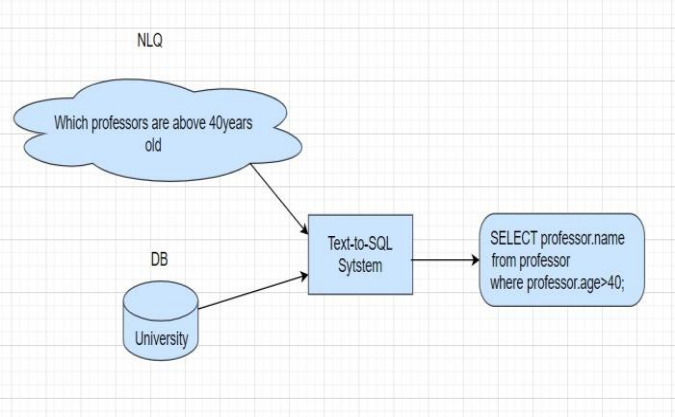

Fig-1

3. DATASETS

A text-to-SQL dataset, also known as a benchmark, comprises pairs of natural language (NL) queries and their corresponding SQL queries, typically defined over one or more databases. In the earlier development of systems, there was a lack of standardized datasets for evaluation. Instead, various systems utilized diverse datasets that incorporated different databases with varying sizes and complexities. This absence of a common dataset hindered a fair cross-system comparison and constrained a comprehensive understanding of the text-to-SQL scenario. Traditional deep learning Text-to-SQL systems rely on a set of queries for training. However, the introduction of WikiSQL in 2017 and Spider in 2018 addressed this issue by providing standardized datasets. These datasets enable the training and evaluation of text-to-SQL systems and facilitate comparisons across different approaches. This shift towards common datasets has significantly contributed to improved system evaluations and a clearer perspective on the state of text-to-SQL technology. Datasets can be divided into two broad categories, Domain specific and Cross-Domain.

3.1.Domain Specific:

Domain specific Text-to-SQL databases are specific to particular domains, examples being IMDB[7], Yelp, and medical data (MIMICSQL). However, these datasets exhibit several drawbacks:

Given their emphasis on a single domain, they are not suitable for comparison against systems that utilize databases from various domains. These datasets tend to be relatively small in size when compared to larger counterparts like Spider and WikiSQL, often containing no more than a thousand examples. These datasets lack a predefined train/dev/test split, which can pose challenges for consistent and standardized model evaluations.

The significance of databases specific to particular domains becomes particularly apparent in scenarios where a practical application would realistically demand a text-to-SQL system to operate with a single database of a specific domain. Given the inherent challenges in achieving generalization capability across diverse domains, these datasets find limited application in more widespread contexts.

3.2. Cross Domain Datasets:

The current emphasis in the domain of text-to-SQL has shifted towards datasets that span multiple domains, broadening the task's applicability by incorporating queries from diverse areas. WikiSQL consists of massive collection over 80,000 pairs of SQL queries and respective natural language questions, and is split into separate training ,development and testing sets which are utilized for development of text-to-SQL systems. Nevertheless, many text-to-SQL systems[41,42] were able to achieve over 90% test accuracy due to the simplistic design of dataset. It does not contain a large number of complex SQL structures such as INTERSECT,JOIN,UNION etc. Additionally, WikiSQL is riddled with ambiguities and erroneous examples that result in degradation of system performance. The Spider dataset has ushered in a transformative approach to database creation. Various iterations, including Spider-SYN, Spider-DK, Spider-CG, and Spider-SSP, have been specifically designed to evaluate particular capabilities of text-to-SQL systems. This marks a significant advancement in the field, offering nuanced assessments and testing scenarios. Ex : spider-SYM is a modified version of original spider questions by replacing schema related terms by their synonyms elevating schema linking challenges. Spider-SSP is specifically designed to assess models' generalization abilities, employing diverse strategies such as substitutions and compositional generalizations.

The CoSQL dataset[11], which is known for being the first extensive cross-domain conversational text-to-SQL collection, is made up of roughly 3,000 conversations, which include over 30,000 dialogue terms and 10,000 linked SQL queries. This dataset simulates scenarios in which natural language is used by annotators who are acting as database users to obtain information from databases. Another noteworthy dataset, CHASE[8], offers a large-scale Chinese text-to-SQL dataset that is context dependent. 17,940 individual questions are matched with SQL queries, and 5,459 question sequences are included. In a recent benchmark known as BIRD [9], which stands for Big Bench for Large-scale Database, complexity is heightened through the inclusion of 12,751 examples involving querying information across 95 large databases. These databases collectively amount to a size of 33.4GB and span 37 professional domains. An interesting feature of BIRD is the introduction of a unique evaluation metric named the Valid Efficiency Score (VES), aimed at assessing the efficiency of the produced SQL queries. Together, these datasets broaden the scope of the text-to-SQL

field, providing dialogue-focused interactions with databases and a range of challenges for research investigation.

4. EVALUATION

4.1. Evaluation metrics:

Evaluating the performance of Text-to-SQL systems is a critical aspect of understanding their effectiveness and practical applicability. It helps identify the accuracy/performance of the system in converting diverse natural language queries into semantically correct SQL expressions. The evaluation of a generated query by a Text-to-SQL system commonly entails comparing it to a ground truth query, often referred to as gold SQL. Evaluation metrics can be classified into three types --string matching, execution matching and manual evaluation.

4.1.1. String matching

Exact match:

Exact String Match[20] is the strictest metric. It demands an exact match between the generated query and the reference gold SQL, right down to every character and a match is found only when generated query is same as ground truth SQL. This includes the order of clauses , aliases and even the formatting used. It is widely used to evaluate the Text to sql systems especially by rule-based and Template-based approaches where a predefined structure is to be followed. However, its strictness in exact string matching can be a drawback. An SQL query can be constructed in ways that differ in syntactical structure but produce semantically equivalent results. This makes Exact String matching unsuitable for deep learning approaches that emphasize flexibility in query construction.

4.1.2. Execution Match:

In contrast to exact-match methods that directly compare structural components or strings in the output, execution match assesses the correctness of a semantic expression based on results obtained after execution against the database. If the results align, the generated query is deemed correct, irrespective of syntactic differences with the gold SQL. This metric is particularly valuable in scenarios where distinct query expressions may produce the equivalent desired output. However, a drawback of this metric is the potential for false positives, where two queries, despite yielding identical results, exhibit differences at a semantic level. For example, when both queries return empty results or when conditional filtering is applied to distinct columns, coincidentally resulting in the same outcome.

4.2. Manual Evaluation:

Manual evaluation involves human evaluators assessing the generated SQL queries to determine how effectively they capture the user's intent. Human evaluation plays a crucial role in identifying the nuanced aspects of semantic equivalence, especially in situations where the execution results of two expressions may differ, yet both are valid in real-world contexts. Consider restaurant-related queries where both **SELECT name FROM restaurants WHERE cuisine = "Italian"** and **SELECT id, name FROM restaurants WHERE cuisine = "Italian"** could be recognized as valid responses when inquiring about Italian cuisine establishments, even though they result in different outputs. Although manual evaluation provides in-depth insights, it is a labour-intensive and time-consuming process, and the subjective nature of evaluators can impact the results. Therefore, manual evaluation is often used along with automatic evaluation metrics for comprehensive evaluation.

5. Literature Survey

Over the past two decades, there has been a considerable progress within the field of text-to-SQL systems through several phases .Text-to-sql systems can be broadly classified into three phases. Foundational phase ,Neural network phase, Language model phase. This section serves as a brief introduction, outlining the key developments that have shaped the text-to-SQL landscape.

5.1. Foundational phase:

In Rule-based approaches ,generation of the sql query is dependent on a predefined set of rules and patterns. Simplistic design and efficiency are its advantages. Initial studies within the field have utilized domain-specific, supporting limited natural language[14], rule-based [23] and template based[22,24,25] methods. Rule-based study employed templates to align natural language sentences directly with string patterns. Utilized a pattern to formalize the syntax tree, aligning it with the syntax analysis tree of the given natural language sentence. However, they also come with numerous drawbacks as the databases grow in complexity and when the task requires handling of intricacies of natural language. They suffer from poor generalizability and applicability. While [12] provided increased domain versatility utilizing supervised models trained across various domains and datasets.

5.2. Neural networks and PLM's phase:

Notable advancements in the field of deep learning [26,27,28] ,have contributed significantly towards the progress of Text-to-sql task. Neural approaches provide increased flexibility in handling complex and varied natural language queries, allowing for better adaptability to diverse user inputs and query structures. State-of-the-art works along this line are[29,30] On the contrary ,these approaches pose a few key challenges such as syntactically incorrect query being generated as the output sequence, picking a wrong column or table names and the assumption that the problem is a simple sequence-to-sequence approach. In response to these challenges, the widespread adoption of the pre-trained language models(PLMs) such as [28,31] has significantly elevated performance across numerous NLP problems including text-to-sql. Prominent studies [29,32,33] have showcased their efficacy in achieving state-of-the-art accuracy on the Spider dataset. Typically, these methods involve the specialized training of models, focusing on tabular data and utilizing NL2SQL pairs as the fundamental training format.

5.3. Large Language Models(LLMs) phase:

A large language model (LLM) is a large-scale language model notable for its ability to achieve general-purpose language understanding and text generation. LLMs acquire these abilities by using massive amounts of data to learn billions of parameters during training and consuming large computational resources during their training and operation. Given the rapid surge in the development of large language models like GPT-3, GPT-4, PALM, and numerous open-source variants, researchers have undertaken investigations into their applicability and effectiveness in present text-to-SQL generation[34,37].Promising results have been showcased by the DIN-SQL [35] and DAIL SQL[36] approaches, both relying on GPT-4. Nevertheless, a drawback of the GPT-4 model lies in its computational cost and the requirement of prompting tokens necessary for these approaches. In contrast, GPT-3.5 based approach C3[38] has exhibited comparable accuracy, effectively addressing the challenges posed by GPT-4-based approaches. A comprehensive survey[37] underscores the notable gap between open-source models and state-of-the-art approaches, revealing the extent to which the former still lags behind. The key advantage of large language model (LLM) based approaches lies in their utilization of zero-shot [39] and few-shot learning principles[40] . A recent study Bird[9] emphasizes on challenging LLM-based state of the art approaches by incorporating databases that simulate real world applications. It has shown that GPT-4 based DIN-SQL[35] outperforms all baseline language models. However, the execution accuracy of DIN-SQL lags behind human capabilities by a significant margin.

6. Conclusion:

In summary, this review paper has emphasized the development and significance of diverse text-to-SQL systems, crucial for facilitating communication between users and databases, particularly for those lacking proficiency in SQL. The paper delves into the evolution of datasets and broad categories of evaluation procedures that gauge the proficiency of these systems in translating natural language queries into database queries. Furthermore, it illuminates the key factors behind the advancements witnessed over the years, outlining the progression of text-to-SQL systems towards the latest state-of-the-art techniques. Despite commendable progress, current methodologies fall short of emulating human performance and capabilities within this domain. Consequently, this field continues to be an active area of research, demanding further exploration to address existing gaps. In the future, LLM-based systems will extend beyond providing SQL

queries and will also have the capability to generate natural language responses by interpreting the execution results derived from the database after running the generated SQL query.

7. References:

[1] Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B. and Zhou, J., 2023. Text-to-sql empowered by large language models: A benchmark evaluation. arXiv preprint arXiv:2308.15363.
[2] J. M. Zelle and R. J. Mooney, "Learning to parse database queries using inductive logic programming," in Proceedings of the national conference on artificial intelligence, 1996.
[3] Zhong, R.Y., Xu, X., Klotz, E. and Newman, S.T., 2017. Intelligent manufacturing in the context of industry 4.0: a review. Engineering, 3(5), pp.616-630.
[4] Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S. and Zhang, Z., 2018. Spider.
[5] Affolter, K., Stockinger, K. and Bernstein, A., 2019. A comparative survey of recent natural language interfaces for databases. The VLDB Journal, 28, pp.793-819.
[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," Proc. of NeurIPS, 2014.
[7] Yaghmazadeh, N., Wang, Y., Dillig, I., Dillig, T.: Sqlizer: query synthesis from natural language. In: PACMPL, pp. 63:1–63:26 (2017)
[8] Guo, J., Si, Z., Wang, Y., Liu, Q., Fan, M., Lou, J.G., Yang, Z. and Liu, T., 2021, August. Chase: A Large-Scale and Pragmatic Chinese Dataset for Cross-Database Context-Dependent Text-to-SQL.
[9] Li, J., Hui, B., Qu, G., Li, B., Yang, J., Li, B., Wang, B., Qin, B., Cao, R., Geng, R. and Huo, N., 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *arXiv preprint arXiv:2305.03111*.
[10] Zhong, V., Xiong, C., Socher, R.: Seq2sql: generating structured queries from natural language using reinforcement learning (2017)
[11] Yu, T., Zhang, R., Er, H.Y., Li, S., Xue, E., Pang, B., Lin, X.V., Tan, Y.C., Shi, T., Li, Z. and Jiang, Y., 2019. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. *arXiv preprint arXiv:1909.05378*.
[12] Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao,Jian-Guang Lou, Ting Liu, and Dongmei Zhang.2019. Towards complex text-to-sql in cross-domain database with intermediate representation.
[13] John M. Zelle and Raymond J. Mooney. 1996. Learn-ing to parse database queries using inductive logic programming. In AAAI.
[14] A. Popescu, A. Armanasu, O. Etzioni, D. Ko, and A.2004, Yates.Modern natural language interfaces to databases: Composing sta-tistical parsing with semantic tractability.
[15] Niculae Stratica, Leila Kosseim, and Bipin C Desai.2005. Using semantic templates for a natural lan-guage interface to the cindi virtual library.
[16] Abbas, S., Khan, M.U., Lee, S.U.-J., Abbas, A., Bashir, A.K.: Areview of nlidb with deep learning: findings, challenges and openissues. IEEE Access. 10, 14927–14945 (2022)Affolter, K., Stockinger, K., Bernstein.
[17] R. C. A. Iacob, F. Brad, E. S. Apostol, C. Truica, I. Hosu, and T. Rebedea. Neural approaches for natural language interfaces to databases: A survey. In D. Scott, N. Bel, and C. Zong, editors, Proceedings of COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 381–395, 2020.
[18] G. Katsogiannis-Meimarakis and G. Koutrika. A survey on deep learning approaches for text-to-sql. VLDB J., 32(4):905–936, 2023.
[19] Deng, N., Chen, Y., Zhang, Y.: Recent advances in text-to-SQL:a survey of what we have and what we expect. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 2166–2187. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (2022)
[20] C. Finegan-Dollak, J. K. Kummerfeld, L. Zhang, K. Ramanathan, S. Sadasivam, R. Zhang, and D. R. Radev. Improving text-to-sql evaluation methodology. In I. Gurevych and Y. Miyao, editors, Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018.
[21] Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., Radev, D.: 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, pp. 3911–3921.
[22] DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin.2021. Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases. Computational Linguistics 47, 2 (2021), 309–332.
[23] Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish R. Mittal, Diptikalyan Saha, and Sankaranarayanan. 2020.
[24] Fu, H., Liu, C., Wu, B., Li, F., Tan, J. and Sun, J., 2023. CatSQL: Towards Real World Natural Language to SQL Applications. Proceedings of the VLDB Endowment, 16(6), pp.1534-1547.
[25] Wonseok Hwang, Jinyeung Yim, Seunghyun Park, and Minjoon Seo. 2019. A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization. arXiv preprint (2019).
[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L.,Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need.In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, pp. 6000–6010. Curran Associates Inc., Red Hook, NY (2017)
[27] Luong, T., Pham, H., Manning, C. D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421. Association for Computational Linguistics, Lisbon (2015).
[28] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.
[29] Scholak, T., Schucher, N. and Bahdanau, D., 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. arXiv preprint arXiv:2109.05093.
[30] Wang, B., Shin, R., Liu, X., Polozov, O. and Richardson, M., 2019. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. arXiv preprint arXiv:1911.04942.
[31] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), pp.5485-5551.
[32] Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, FeiHuang, Wenyu Du, Luo Si, and Yongbin Li. 2023. Graphix-T5: Mixing Pre-trained Transformers with Graph-Aware Layers for Text-to-SQL Parsing. In 37th AAAI Conference on Artificial Intelligence. 13076–13084.
[33] B. Hui, R. Geng, L. Wang,B. Qin, B. Li, J. Sun, andY. Li, "S 2 sql: Injecting syntax to question-schema in-teraction graph encoder for text-to-sql parsers," ArXivpreprint, 2022.
[34] Aiwei Liu, Xuming Hu, Lijie Wen, and Philip SYu. 2023. A comprehensive evaluation of chat-gpt's zero-shot text-to-sql capability.
[35] M. Pourreza and D. Rafiei. DIN-SQL: decomposed in-context learning of text-to-sql with self-correction. 2023.
[36] Zhang, H., Cao, R., Chen, L., Xu, H. and Yu, K., 2023. ACT-SQL: In-Context Learning for Text-to-SQL with Automatically-Generated Chain-of-Thought.
[37] Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B. and Zhou, J., 2023. Text-to-sql empowered by large language models: A benchmark evaluation. arXiv preprint arXiv:2308.15363.
[38] Dong, X., Zhang, C., Ge, Y., Mao, Y., Gao, Y., Lin, J. and Lou, D., 2023. C3: Zero-shot Text-to-SQL with ChatGPT.
[39] Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. and Iwasawa, Y., 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35, pp.22199-22213.
[40] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. Advances in neural information processing systems, 33, pp.1877-1901.
[41] Xu, K., Wang, Y., Wang, Y., Wen, Z. and Dong, Y., 2021. Sead: End-to-end text-to-sql generation with schema-aware denoising. *arXiv preprint arXiv:2105.07911*.
[42] Hui, B., Shi, X., Geng, R., Li, B., Li, Y., Sun, J. and Zhu, X., 2021. Improving text-to-sql with schema dependency learning. *arXiv preprint arXiv:2103.04399*.