Towards Robustness of Large Language Models on Text-to-SQL Task: An Adversarial and Cross-Domain Investigation

Weixu Zhang, Yu Wang, and Ming Fan

Xi'an Jiaotong University, Xi'an, 710049, China
weixu_zhang,uyleewang@stu.xjtu.edu.cn
mingfan@mail.xjtu.edu.cn

Abstract. Recent advances in large language models (LLMs) like ChatGPT have led to impressive results on various natural language processing (NLP) challenges including text-to-SQL task, which aims to automatically generate SQL queries from natural language questions. However, these language models are still subject to vulnerabilities such as adversarial attacks, domain shift and lack of robustness, which can greatly affect their performance and reliability. In this paper, we conduct a comprehensive evaluation of large language models, such as ChatGPT, on their robustness in text-to-SQL tasks. We assess the impact of adversarial and domain generalization perturbations on LLMs using seven datasets, five of which are popular robustness evaluation benchmarks for text-to-SQL tasks and two are synthetic adversarial datasets generated by ChatGPT. Our experiments show that while LLMs exhibit promise as zero-shot text-to-SQL parsers, their performances degrade under adversarial and domain generalization perturbations, with varying degrees of robustness depending on the type and level of perturbations applied. We also explore the impact of usage-related factors such as prompt design on the performance and robustness of LLMs. Our study provides insights into the limitations and potential directions for future research to enhance the performance and robustness of LLMs on text-to-SQL and other NLP tasks.

Keywords: Large language model · ChatGPT · text-to-SQL · Robustness · Adversarial attacks.

1 Introduction

Text-to-SQL is a natural language processing task that aims to automatically generate structured SQL queries from natural language questions[4]. This task has been an active research area in natural language processing field with a wide range of applications, including database querying, question answering, and data retrieval. Although existing models have achieved impressive performance on many public benchmarks[20,11], research work has found that text-to-SQL models exhibit a lack of robustness under various conditions including adversarial

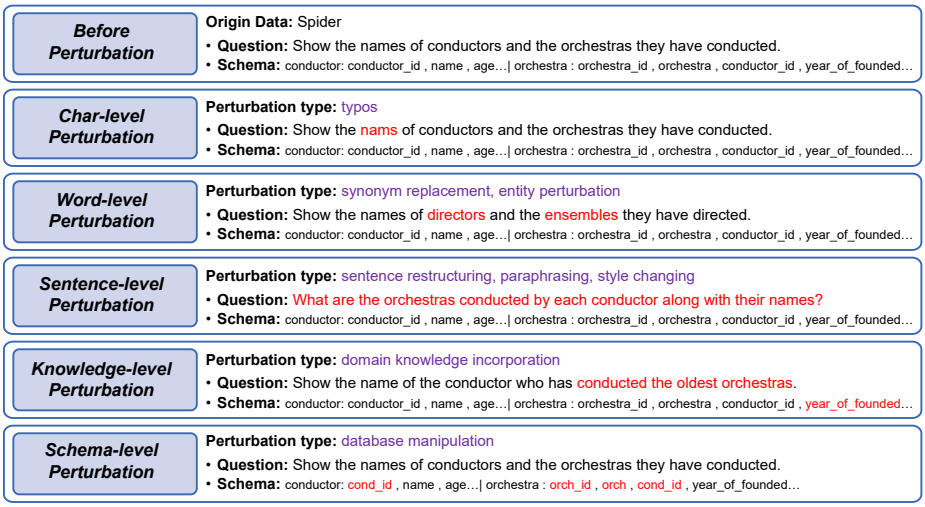

Fig. 1. An example in Spider dataset with different levels of perturbations.

attacks and domain shifts[6,7], where small perturbations added to the input text can severely deteriorate the performance of the victim model.

Figure 1 provides examples that showcase how state-of-the-art text-to-SQL models can be vulnerable to perturbations, which can come in a variety of forms and levels. These perspectives of perturbation include adversarial and domain generalization. Adversarial perturbations can be introduced into input data at different levels (character, word, or sentence) by creating typos, synonym replacement, entity perturbation, or sentence restructuring[3]. Additionally, adversarial examples can be generated by modifying the database, such as synonym or abbreviation replacement[13] on the database schema. On the other hand, the cross-domain perspective evaluates the model's ability to generalize to significantly different domains, and knowledge-level perturbations can be introduced by incorporating domain-specific knowledge[7].

Large language models (LLMs), such as ChatGPT, have received increasing attention in recent months due to their impressive performance on wide range of natural language processing tasks[16]. However, the robustness of these models against adversarial attacks and cross-domain data remains a challenge. While previous studies have evaluated the robustness of LLMs on other NLP tasks, such as text classification and sentiment analysis[19], their robustness on text-to-SQL tasks has not been thoroughly investigated. Given the ever-increasing use of LLMs in real-world applications, it is essential to evaluate the robustness of these models on this task and understand their limitations to identify potential areas for improvement.

In this work, we conduct a comprehensive evaluation of LLMs on its adversarial and domain generalization robustness on text-to-SQL tasks. We aim to explore the extent to which the performances of LLMs are affected when tested

on adversarial and domain generalization data. We used 7 datasets including 5 popular robustness evaluation benchmark of text-to-SQL task and 2 dataset generated by ChatGPT to cover all levels of perturbations (character, word, sentence, schema, etc.) and overall 20k+ test examples. We also selected several SOTA text-to-SQL models to compare with LLMs.

Moreover, the performance of LLMs is known to depend heavily on its usage, including prompt design, input formatting, parameter setting and fine-tuning[21,12]. In this paper, we investigate the impact of usage-related factors such as prompt designs and temperature setting on the performance and robustness of LLMs in the text-to-SQL task. Our study aims to provide insights into the limitations of current LLMs and potential directions for future research to enhance their performance and robustness on text-to-SQL and other NLP tasks. Our key findings and insights are:

- LLMs like ChatGPT show promise as zero-shot text2sql parsers, but they are not yet expert compared to carefully designed and fine-tuned models. However, their performance has been improving with version updates.
- The performance of LLMs degrades in the face of perturbations of adversarial and domain generalization. However, LLMs are also more robust to adversarial examples than domain generalization examples.
- LLMs exhibit varying degrees of robustness depending on the type and level of perturbations applied, with NLP perturbations being easier to handle than DB perturbations.
- LLMs perform better on examples generated by LLMs themselves than on human-created datasets,showing the potential benefits of these models in generating synthetic data for training.
- Usage-related factors such as prompt design and temperature settings influence the performance of LLMs on text2sql tasks. Providing additional information in prompts such as CREATE TABLE commands and more examples improves the performance of LLMs on text-to-SQL tasks.

2 Background

2.1 Text-to-SQL problem

The task of converting natural language into structured query language (SQL), which is commonly referred to as text-to-SQL problem, has been studied extensively in the NLP community[4]. Approaches based on rule, pattern or grammar have been proposed during early years. With the advent of machine learning techniques, researchers have explored statistical models to learn the mapping between natural language and SQL. The release of Spider leaderboard in 2018[22] has sparked a strong interest in approaches using advanced neural networks[18]. With the prevalence of pretrained language models, most recent works tend to build transformer-based parsers with pretrained models like T5[15] and Bart[10]. The use of LLMs, such as GPT-3[1] and ChatGPT, has also shown impressive results on text-to-SQL task in recent months.

2.2 Large Language Models and Prompting

Large Language Models (LLMs) have garnered significant attention in recent research for their ability on various downstream NLP tasks[9].These models generate high-quality responses by training on massive amounts of text data[15,10]. Improved prompting has been identified as a key factor in effectively applying the information contained in LLMs to target tasks. Carefully designed prompts can provide contextual information that guides the LLMs to generate intermediate reasoning steps and high-quality responses through the establishment of a Chain-of-Thought. One most notable example is ChatGPT, a variant of the GPT (Generative Pre-trained Transformer)[1] model family specifically designed to generate human-like text, which has shown impressive performances on various NLP tasks, making it an attractive candidate for text2sql task.

2.3 Robustness

The robustness of natural language processing models, including those for text-to-SQL, is a critical concern for their practical deployment. Although existing Text-to-SQL parsers have achieved good performance on many public datasets, they were recently found to be vulnerable to adversarial attacks and domain generalization[6,13]. Perturbations can come in a variety of forms sides, including perturbations on natural language question and database schema[5]. Adversarial examples can also be generated at different levels(character, word, or sentence) by creating typos, synonym replacement, or sentence restructuring. In addition to adversarial attacks, domain generalization is another challenge that can affect the performance of parsers. This problem arises when models encounter rarely observed domain-specific knowledge, which can lead to incorrect SQL queries in real-world applications[7].

3 Datasets and Tasks

In this work, we utilize the widely used Spider dataset[22] as our baseline. To assess robustness of text-to-SQL models against a range of challenging inputs, we incorporate several evaluation benchmarks curated from the Spider development dataset that encompass both adversarial and domain-generalization perspectives. Table 1 presents statistics of evaluation datasets we use.

Adversarial Dataset We evaluate the robustness of LLMs on text-to-SQL tasks against adversarial inputs using a combination of publicly available datasets and synthetic adversarial examples generated by the gpt-3.5-turbo(ChatGPT) model. The three publicly available datasets that we use are Spider-Syn, Spider-Realistic, and Dr.Spider. **Spider-Syn**[6] includes questions that have been perturbed on a word level using synonym replacement to test the model's ability to handle variations in expressions for the same meaning.

Table 1. Statistics of robustness evaluation datasets in this paper. ADV_T and ADV_S are adversarial datasets curated from Spider with gpt-3.5-turbo on perturbations of typo creating and style changing. Level C, W, S ,D, K represents perturbations in the levels of character, word, sentence, database, knowledge, respectively.

Type	Dataset	Size	Data manipulation	Level
Standard	Spider	1034	no manipulation	-
Adversarial	ADV_T	1034	typo creating	C
	Spider-Syn	1034	synonym replacement	W
	Spider-Realistic	508	paraphrasing	S
	ADV_S	1034	style changing	S
	Dr.Spider	15k	multitype perturbation	W, S, D
Cross-domain	Spider-DK	535	knowledge incorporation	K

Spider-Realistic[5] includes sentence-level perturbations of removing explicitly mentioned column names in natural language questions. **Dr.Spider**[3] is a comprehensive diagnostic benchmark containing 15,000 perturbed examples covering multiple types of perturbations from three perspectives: database, natural language questions, and SQL.

In addition to these three datasets, we use synthetic adversarial examples generated by ChatGPT to evaluate the robustness of LLMs against two specific types of perturbations not covered by the aforementioned benchmarks. The first type, $\mathbf{ADV}_T$, includes typo-creating perturbations such as adding, deleting, or altering a character in a word, designed to test the model's robustness against character-level changes. The other type, $\mathbf{ADV}_S$, includes perturbations where the writing style of the question is altered, such as changing the tone or language register, to test the model's robustness against stylistic changes.

Domain Generalization Dataset To evaluate the robustness of LLMs on text-to-SQL under cross-domain conditions, we incorporate the **Spider-DK** dataset[7]. Spider-DK is a human-curated dataset based on Spider for evaluating the generalization ability of text-to-SQL models across different domains. It contains 535 samples incorporating domain information to paraphrase questions to assess performances of LLMs where domain-specific knowledge is essential for accurate translations.

4 Experiments

4.1 Experiment Setup

Baselines To evaluate the robustness of the LLMs on text2sql tasks, we compared their performance against several state-of-the-art baseline

Table 2. Exact match accuracy (EM) and execution accuracy (EX)(%) results on Spider, Spider-DK, Spider-Syn, and Spider-Realistic.

Model	Spider		Spider-DK		Spider-Syn		Spider-Realistic	
	EM	**EX**	**EM**	**EX**	**EM**	**EX**	**EM**	**EX**
Finetuned								
RAT-SQL + BERT	-	-	40.9	-	48.2	-	58.1	62.1
RAT-SQL + GRAPPA	73.4	-	38.5	-	49.1	-	59.3	-
LGESQL + ELECTRA	75.1	-	48.4	-	64.6	-	69.2	-
LGESQL + ELECTRA + SUN	-	-	52.7	-	66.9	-	70.9	-
TKK-3B	-	-	-	-	63.0	68.2	68.5	71.1
T5-3B	71.5	74.4	-	-	59.4	65.3	63.2	65.0
T5-3B + PICARD	75.5	79.3	-	-	-	-	68.7	71.4
RASAT + PICARD	75.3	80.5	-	-	-	-	69.7	71.9
RESDSQL-3B + NatSQL	**80.5**	**84.1**	**53.3**	**66**	**69.1**	**76.9**	**77.4**	**81.9**
$Inference - only$								
text-davinci-002	6.0	12.5	5.5	11.4	4.2	10.5	5.8	11.6
text-davinci-003	31.9	55.6	30.7	53.8	22.8	43.3	30.7	52.0
gpt-3.5-turbo	**46.6**	**71.4**	**41.5**	**59.1**	**38.9**	**61.7**	**41.1**	**64.6**

models. We tested the LLMs accessible via the OpenAI API, including **ChatGPT**(gpt-3.5-turbo), **GPT-3** (text-davinci-002, text-davinci-003)[1], and **Codex**. GPT-3 is trained on diverse sources of text from the internet, Codex is further fine-tuned on code from GitHub, while ChatGPT gained further performance boosts from RLHF(reinforcement learning from human feedback). We conducted zero-shot evaluations of these models on the previously mentioned text2sql benchmarks. We also compared their performances with several state-of-the-art text2sql baseline models fine-tuned from the Spider training set. They have been previously reported as SOTA on Spider benchmark which use various techniques including attention mechanisms(**RATSQL**[18], **RASAT**[14]), graph-based approaches(**LGESQL**[2]), pretained language models(**T5-family**[20]), process decoupling(**TKK[8]**), carefully designed encoders and decoders(**RESDSQL**[11],**PICARD**[17]) to address generalization and robustness challenges.

Evaluation metric To evaluate the performance of the Text-to-SQL parser, we used two metrics: Exact set match accuracy (EM) and Execution accuracy (EX)[22]. EM measures the percentage of queries for which the generated SQL query exactly matches the gold SQL query, while EX compares the execution results of the predicted and gold SQL queries on databases, with the percentage of queries that produce correct results.

EM and EX are essential in measuring the parser's ability to generate semantically complete and accurate SQL queries, and produce expected results. However, EM lacks flexibility and can produce false-negative examples, whereas the EX metric is more sensitive to the generated values.

Table 3. Data statistics and the execution (EX) accuracy of SOTA text-to-SQL models on the original Spider development set (Spider-dev) and Dr.Spider(%).

	Perturbation	sample	RATSQL	T5-3B	PICARD	RESDSQL	CodeX	GPT-3.5-turbo
	Spider-dev	1,034	72.8	71.7	79.3	**84.1**	67	71.4
DB	Schema-synonym	2,619	45.4	41.6	56.5	**68.3**	62	53.7
	Schema-abbreviation	2,853	44.2	50.7	64.7	**70**	68.6	64
	DBcontent-equivalence	382	12	36.4	43.7	40.1	**51.6**	41.9
	Avergae	-	33.9	42.9	55	59.5	**60.7**	53.2
NLQ	Keyword-synonym	953	53.7	60.3	66.3	**72.4**	55.5	57.6
	Keyword-carrier	399	81	76.9	82.7	83.5	**85.2**	78.4
	Column-synonym	563	42.6	46.5	57.2	**63.1**	54.7	47.4
	Column-carrier	579	58	59.6	64.9	63.9	51.1	**67.4**
	Column-attribute	119	42.9	52.1	56.3	**71.4**	46.2	61.3
	Column-value	304	52.6	50	69.4	**76.6**	71.4	56.3
	Value-synonym	506	18.6	35.8	53	53.2	**59.9**	46.2
	Multitype	1,351	39.7	47	57.1	**60.7**	53.7	52.9
	Others	2,819	67.2	66	78.3	**79**	69.7	65.1
	Average	-	50.7	54.9	65	**69.3**	60.8	59.2
SQL	Comparison	178	59.6	60.1	68	**82**	66.9	69.1
	Sort-order	192	68.2	73.4	74.5	**85.4**	57.8	60.9
	NonDB-number	131	58.8	82.4	77.1	85.5	**89.3**	89.3
	DB-text	911	51.2	52.1	65.1	**74.3**	72.4	65.5
	DB-number	410	74.4	79.3	85.1	**88.8**	79.3	77.1
	Average	-	62.4	69.5	74	**83.2**	73.1	72.4
All		-	51.2	57.1	65.9	**71.7**	64.4	61.6

4.2 Results

RQ1: How effective and robust are LLMs in zero-shot text2sql parsing compared to fine-tuned text2sql models?

Table 2 presents EM and EX results on Spider, Spider-DK, Spider-Syn, and Spider-Realistic. Overall, our experiments demonstrate that large language models (LLMs) like ChatGPT show promise as zero-shot text2sql parsers, but are not yet expert compared to carefully designed and fine-tuned text2sql models. However, performances of LLMs have been improving with version updates, suggesting that they have potential to gain expert-level performance in future.

Additionally, we found that LLMs achieved higher results on EX accuracy than EM accuracy. This suggests that LLMs are better at producing SQL queries that produce the expected results rather than generating the SQL queries that match the exact components of the gold SQL. Furthermore, our experiments showed that LLMs are more robust to adversarial examples (Spider-Syn, Spider-Realistic) than domain generalization (Spider-DK) examples, but their performance degrades in the face of perturbations of both types.

Table 4. Statistics and evaluation results on adversarial test data generated by gpt-3.5-turbo

Data	Size	EM	EX	Example utterance
Spider	1034	46.6	71.4	*What is the total number of* ***singers****?*
Spider-Syn	1034	38.9	61.7	*What is the total number of musicians?*
ADV_T	1034	46.5	64.9	*What is teh total number of sngers?*
ADV_S	2068	40.0	64.5	*Show me the count of all the singers.*

RQ2: What is the impact of adversarial attacks with different levels of perturbation on the performance of LLMs?

Table 3 presents more detailed results of models on the Dr. Spider benchmark under different levels and sides of perturbations. Our findings indicate that LLMs exhibit varying degrees of robustness depending on the type and level of perturbations applied.

Specifically, we found that LLMs are generally more robust to natural language processing (NLP) perturbations, such as synonym replacements and paraphrasing, than database (DB) perturbations, such as schema synonym and table content equivalence. They are found more robust to low-level perturbations, such as a small number of surface-level word changes, and show lower robustness when facing high-level perturbations, such as significant changes in data structure.

We also evaluated the performance of large language models (LLMs) on additional test examples generated by ChatGPT using two types of perturbations: typo and style changing. These additional test examples were used to investigate the ability of LLMs to handle character-level and style-level perturbations as well as synthetic data generated by LLMs. Statistics of test data and evaluation results are presented in Table 4.

RQ3: What is the role of LLMs in generating synthetic data for text2sql tasks?

Our results in Table 4 indicate that LLMs perform better on the examples generated by LLMs themselves than on human-created datasets, showing the potential benefits of these models in generating synthetic data for training. Specifically, we observed that LLMs could better handle and correct typos introduced by ChatGPT, suggesting that they have the potential to improve their own generated data quality.

4.3 Case Study

Table 5 presents a case study using an original example fetched from the Spider benchmark to further illustrate the performance of large language models on the text2sql task under different levels of perturbations including typos, synonym replacement and sentence restructuring.

Table 5. Examples of different types of perturbations applied to a natural language question (NL) from the Spider benchmark, along with the associated ground truth SQL (Gold) and predicted SQL queries (Pred) made by the GPT-3.5-turbo model using question and schema prompts.

NL	**How many different series and contents are listed in the TV Channel table?**
Gold	`SELECT count(DISTINCT series_name) , count(DISTINCT content) FROM TV_Channel`
Pred	`SELECT COUNT(DISTINCT series_name), COUNT(DISTINCT content) FROM tv_channel`
NL	How many different serias and contents are listed in the TV Channel table?
Gold	`SELECT count(DISTINCT series_name) , count(DISTINCT content) FROM TV_Channel`
Pred	`SELECT COUNT(DISTINCT series_name), COUNT(DISTINCT content) FROM tv_channel`
NL	How many different serial and contents are listed in the TV Channel table?
Gold	`SELECT count(DISTINCT series_name) , count(DISTINCT content) FROM TV_Channel`
Pred	`SELECT COUNT(DISTINCT content) FROM tv_channel`
NL	Count the different series and contents listed in the TV Channel table.
Gold	`SELECT count(DISTINCT series_name) , count(DISTINCT content) FROM TV_Channel`
Pred	`SELECT COUNT(*) FROM tv_channel`

This example indicates that the large language models like ChatGPT can produce reasonably accurate SQL queries under low-level perturbations such as typos, but struggle to handle high-level perturbations that involve synonym replacement or sentence paraphrasing of the original question. Specifically, we observed that the large language models could omit perturbed entities or misunderstand the intent of the perturbed question, leading to inaccurate or partial SQL queries. However, it is still noteworthy that the large language models are able to generate readable and partially correct SQL queries in the presence of the high-level perturbation, which suggests their potential for handling diverse types of inputs.

5 Discussion

5.1 The Effect of Temperature

The temperature setting is a crucial hyperparameter for LLMs. It controls the degree of randomness in the generated output and affects the trade-off between accuracy and diversity of the generated text. Figure 2 shows the performance of LLMs on text2sql task under different temperature settings.

Specifically, the result suggests that for both ChatGPT and Text-Davinci-003, higher temperature settings tend to degrade the model's performance on the text2sql task. This can be attributed to the increased level of randomness in the generated output, which can lead to a higher likelihood of generating incorrect or nonsensical queries. According to our findings, controlling the temperature between 0 and 0.4 can lead to a better balance between accuracy and diversity. We also found that gpt-3.5-turbo is more robust and less affected by temperature settings compared to Text-Davinci-003.

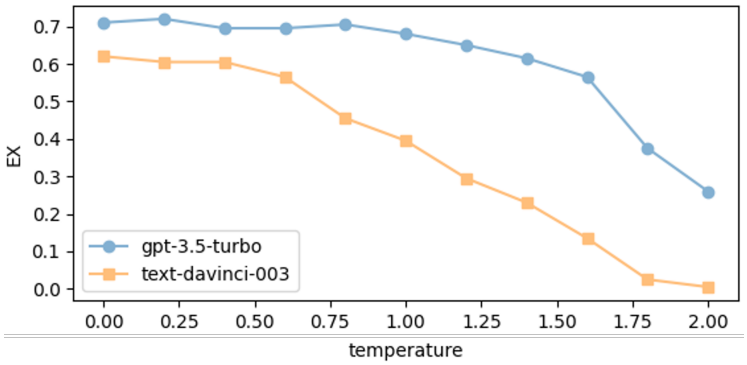

Fig. 2. The relationship between temperature and performance of LLMs on text2sql task evaluated by EX on Spider development set.

5.2 The Effect of Prompt Design

The design of the prompts plays a critical role in the performance of LLMs on text2sql tasks. In our study, we explored five prompt structures: (1) question only, providing no information about the database, (2) question and serialized schema as in T5 inputs, (3) question and schema formatted as API documentation, (4) question and CREATE TABLE commands for the related database, including column types and foreign key declarations, (5) "one-shot" and "five-shot" prompts that included the question, serialized schema, and one or five random examples from the Spider training set.

The results are shown in Figure 3. Our analysis reveals that providing additional information in prompts can significantly improve the performance of LLMs on text-to-SQL tasks. We found that using a more comprehensive schema prompt, such as providing CREATE TABLE commands, can enhance the model's ability to generate accurate SQL queries. Moreover, using additional examples from the training set as part of the prompt, as in the "one-shot" and "five-shot" prompts, can also help the model better understand the structure and relationships of the database and generate more accurate queries.

6 Conclusion

In this study, we investigated the robustness of large language models on Text-to-SQL tasks. Our findings suggest that while LLMs can be good zero-shot text2SQL parsers, but they face challenges in handling adversarial and domain generalization examples with varying degrees of robustness depending on the type and level of perturbations applied. Moreover, the performance of LLMs is influenced by usage-related factors, such as prompt design and temperature settings. Our study provides insights into the strengths and weaknesses of LLMs on Text-to-SQL tasks and suggests future research directions for improving their performance and robustness.

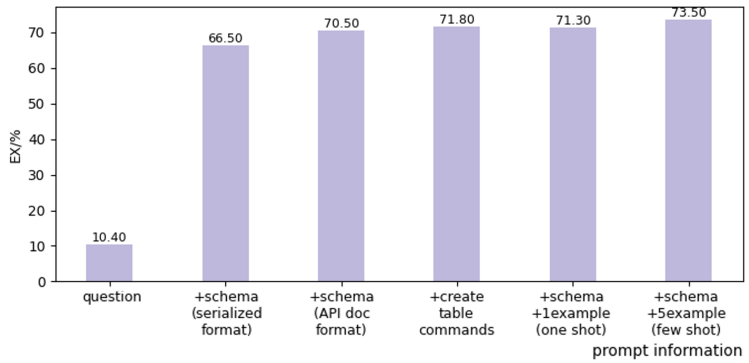

Fig. 3. Performance of LLMs with different prompt information evaluated by EX on Spider development set.

Acknowledgements This work was supported by National Key R&D Program of China (2022YFB2703500), National Natural Science Foundation of China (62232014, 62293501, 62272377, 62293502, 72241433, 61721002, 62032010, 62002280), the Fundamental Research Funds for the Central Universities, CCF-AFSG Research Fund , China Postdoctoral Science Foundation (2020M683507, 2019TQ0251, 2020M673439), and Young Talent Fund of Association for Science and Technology in Shaanxi, China.

References

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
2. Cao, R., Chen, L., Chen, Z., Zhao, Y., Zhu, S., Yu, K.: Lgesql: Line graph enhanced text-to-sql model with mixed local and non-local relations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 2541–2555 (2021)
3. Chang, S., Wang, J., Dong, M., Pan, L., Zhu, H., Li, A.H., Lan, W., Zhang, S., Jiang, J., Lilien, J., Ash, S., Wang, W.Y., Wang, Z., Castelli, V., et al.: Dr.spider: A diagnostic evaluation benchmark towards text-to-sql robustness (2023)
4. 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 (2022)
5. Deng, X., Hassan, A., Meek, C., Polozov, O., Sun, H., Richardson, M.: Structure-grounded pretraining for text-to-sql. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1337–1350 (2021)
6. Gan, Y., Chen, X., Huang, Q., Purver, M., Woodward, J.R., Xie, J., Huang, P.: Towards robustness of text-to-sql models against synonym substitution. In: Proceedings of the 59th Annual Meeting of the Association for Computational

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 2505–2515 (2021)
7. Gan, Y., Chen, X., Purver, M.: Exploring underexplored limitations of cross-domain text-to-sql generalization. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 8926–8931 (2021)
8. Gao, C., Li, B., Zhang, W., et al.: Towards generalizable and robust text-to-sql parsing. arXiv preprint arXiv:2210.12674 (2022)
9. Jiang, Z., Xu, F.F., et al.: How can we know what language models know? Transactions of the Association for Computational Linguistics **8**, 423–438 (2020)
10. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880 (2020)
11. Li, H., Zhang, J., Li, C., Chen, H.: Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql (2023)
12. Peng, K., Ding, L., Zhong, Q., Shen, L., Liu, X., Zhang, M., Ouyang, Y., Tao, D.: Towards making the most of chatgpt for machine translation (2023)
13. Pi, X., Wang, B., Gao, Y., Guo, J., Li, Z., Lou, J.G.: Towards robustness of text-to-sql models against natural and realistic adversarial table perturbation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2007–2022 (2022)
14. Qi, J., Tang, J., He, Z., Wan, X., Zhou, C., Wang, X., Zhang, Q., Lin, Z.: Rasat: Integrating relational structures into pretrained seq2seq model for text-to-sql. arXiv preprint arXiv:2205.06983 (2022)
15. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020), http://jmlr.org/papers/v21/20-074.html
16. Rajkumar, N., Li, R., Bahdanau, D.: Evaluating the text-to-sql capabilities of large language models (2022)
17. Scholak, T., Schucher, N., Bahdanau, D.: Picard: Parsing incrementally for constrained auto-regressive decoding from language models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 9895–9901 (2021)
18. Wang, B., Shin, R., Liu, X., et al.: Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7567–7578 (2020)
19. Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Huang, H., Ye, W., Geng, X., Jiao, B., Zhang, Y., Xie, X.: On the robustness of chatgpt: An adversarial and out-of-distribution perspective (2023)
20. Xie, T., Wu, C.H., Shi, P., Zhong, R., Scholak, T., Yasunaga, M., Wu, C.S., Zhong, M., Yin, P., Wang, S.I., et al.: Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. arXiv preprint arXiv:2201.05966 (2022)
21. Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., Liu, Z., et al.: Mm-react: Prompting chatgpt for multimodal reasoning and action (2023)
22. Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., et al.: 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 (2018)