sonrobok4 at Iberlef 2025 - PRESTA: Leveraging LLMs for Text-to-Python Question Answering over Tabular Data in Spanish

Nguyen Minh Son[1,2,*], Dang Van Thin[1,2]

[1]University of Information Technology-VNUHCM, Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This paper presents our contribution to the PRESTA shared task on Question Answering over Tabular Data in Spanish. We explore the capabilities of large language models (LLMs) for text-to-code generation, focusing on text-to-Python approaches to handle diverse question types. Our method employs a multi-prompt strategy that emphasizes structured table understanding, language-aware prompt construction. We investigate the effectiveness of zero-shot prompting using cutting-edge models such as GPT-4o-mini, DeepSeek-V3, and DeepSeek-R1. Our experiments aim to assess the Python code generation for tabular QA, as well as the robustness of LLMs in handling multilingual and domain-specific tabular contexts. Our approach achieved the highest accuracy among competitors, reaching 87%.

Keywords

Question Answering, Tabular Data, Text-to-Code, Text-to-Python, Large Language Models, Zero-Shot Prompting, Spanish Language, Prompt Engineering,

1. Introduction

The growing interest in question answering (QA) over structured data has led to significant advancements in understanding and reasoning over tabular formats. Although much of this progress has focused on English language datasets, the PRESTA [1] shared task at IberLEF 2025 [2] addresses a critical gap by introducing DataBenchSPA [3], the first large-scale benchmark specifically designed for Spanish QA over tabular data. This initiative opens new opportunities to evaluate and improve QA systems in multilingual and domain-specific contexts. DataBenchSPA comprises diverse real-world tables with varying row and column counts, encompassing a wide range of data types such as numerical, categorical, boolean, and list values. The task challenges participants to build systems that can interpret natural language questions and accurately return answers from the corresponding tables.

Motivated by these questions, we focus our efforts on exploring LLM-based text-to-code generation as a practical solution for this task. In particular, we investigate prompting strategies that encourage structured table understanding while accounting for the linguistic characteristics of Spanish. Through careful experimentation and system design, we aim to highlight the potential of prompt engineering techniques in handling real-world QA over tabular data.

2. Related Work

Question Answering on tabular data represents a critical area of NLP research. Early benchmark datasets including WikiSQL [4], Spider [5], and TabFact [6] primarily focused on English-language evaluation. The development of **databenchSPA** [3], a comprehensive Spanish tabular QA dataset, significantly

IberLEF 2025, September 2025, Zaragoza, Spain
*Corresponding author.
22521254@gm.uit.edu.vn (N. M. Son); thindv@uit.edu.vn (D. V. Thin)
https://nlp.uit.edu.vn/ (D. V. Thin)
0000-0001-8340-1405 (D. V. Thin)

expanded evaluation capabilities by introducing a non-English benchmark, thereby underscoring the importance of multilingual approaches in table-based QA.

Recent advances have increasingly leveraged **Large Language Models (LLMs)** for table QA tasks. Notable approaches include **Chain-of-Table** [7], which dynamically evolves table representations during reasoning, and **Tree-of-Table** [8] that employs hierarchical structures for large-scale table understanding. The DataFrame QA framework introduces a novel method for table question answering without raw data exposure by generating executable pandas queries. Meanwhile, **Table-Critic** [9] demonstrates the effectiveness of multi-agent systems for collaborative table reasoning.

Preliminary experiments with text-to-python conversion on the **databenchSPA** dataset using small open-source models like Mistral [10] and DeepSeek-Coder [11] have revealed promising results while highlighting significant opportunities for further improvement.

3. Methods

Our system employs a Text-to-Code approach utilizes private large language models (LLMs) with cost-awareness to answer questions over tabular data. The architecture comprises several key components: data preprocessing, table context preparation, code generation, an error-correction loop to handle invalid code, and answer synthesis. Each component is described in detail in the following sections. The illustration for our method is in Figure 1

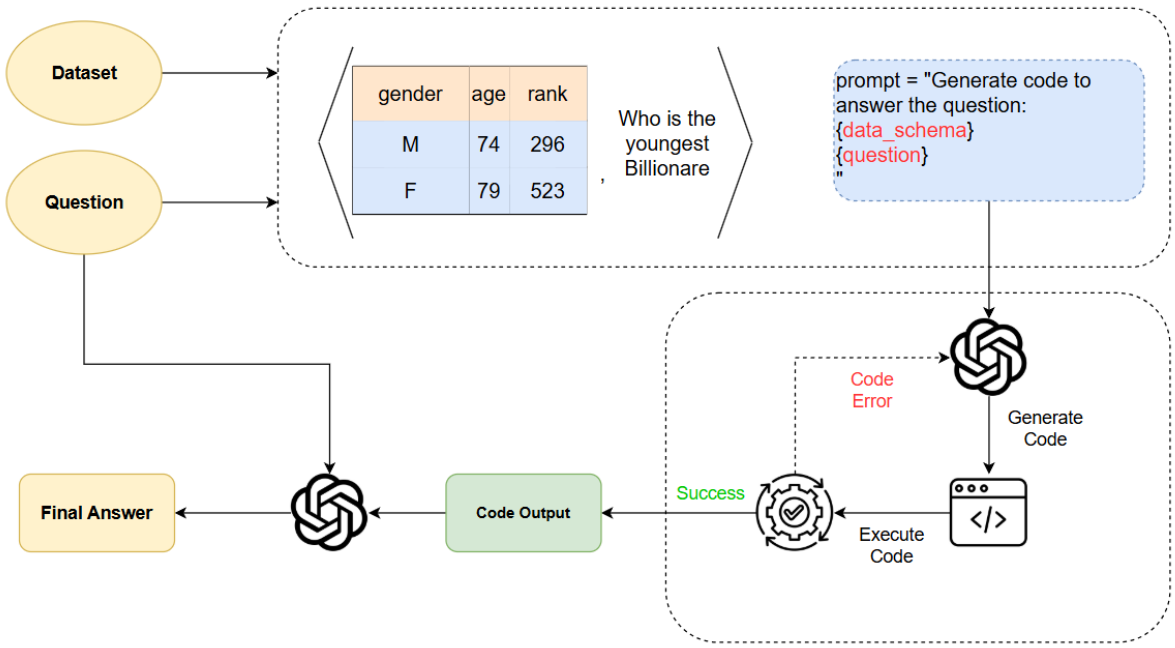

Figure 1: An overview of our system, including code generation, code correction, and answer synthesis.

3.1. Data Preprocessing

An analysis of the dataset revealed that many tables contain columns with a high percentage of null values. Given that each dataset can have approximately 170 columns, it is essential to perform thorough data cleaning by removing unnecessary or sparse columns. This step reduces noise and prevents the large language model (LLM) from being overwhelmed by irrelevant or excessive data during processing.

3.2. Table Context Preparation

During system development, we observed that the type and amount of table information provided to the LLM significantly affect its performance. Specifically, supplying only the first few rows versus including additional metadata such as column names and data types results in notable differences. We provide the first five rows, carefully selected to capture the most unique values per column to maximize the representation of special cases. Additionally, we supply a dictionary containing column names alongside their corresponding data types to enhance the LLM's understanding of the table schema.

3.3. Code Generation

We use zero-shot prompting strategy to generate executable code in Python, depending on the question context. The generated code is then executed on the provided table to produce an intermediate result for the answer synthesis stage. If the execution result is excessively long or malformed, we return `None` to indicate a failure in the code generation logic.

3.4. Correction Loop

To address potential execution errors in the generated code, we implement a correction loop that attempts to revise the code up to five times. In each iteration, the LLM receives the previous code, the error message, and relevant table information to generate a corrected version. This mechanism improves robustness by allowing recovery from common syntax and logic errors.

3.5. Answer Synthesis

The competition defines a constrained set of acceptable output formats, making an answer synthesis module essential for converting raw code output into valid final answers. Based on the type of question and output, the system formats responses into one of the following categories:

- **Boolean**: Valid values include `True/False`, `Yes/No`, or `Y/N` (case-insensitive).
- **Category**: A value or substring from a single cell in the dataset.
- **Number**: A numeric value, potentially derived from calculations such as average, maximum, or minimum.
- **List[category]**: A fixed-length list of categorical values (e.g., `['cat', 'dog']`). The question wording determines whether uniqueness or duplicates are expected.
- **List[number]**: A fixed-length list of numerical values, formatted similarly to `List[category]`.

4. Results and Discussion

4.1. Development Phase

Our experimental results on the development set are summarized in Table **??**. We categorize the experiments into three distinct groups:

1. **Table Input Format and Size:** This group investigates how different input representations affect performance. Specifically, we vary whether the table rows are provided to the LLM in plain string, Markdown, or JSON format. Additionally, we vary the number of rows given: the base case includes 2 rows, while the extended input includes 5 rows.
2. **Prompt Strategy:** We explore the impact of different prompting techniques, including zero-shot, CoT [12], and role-play [13]. These strategies aim to compare LLMs performance across different prompts.
3. **Language Handling:** Since the original dataset is in Spanish, we investigate whether language affects LLM performance. We compare three approaches: translating only the question into English, translating both the question and column headers, and using Spanish prompts.

Table 1
Accuracy of DeepSeek-V3 on different experimental settings.

Group	Experiments Setting	Accuracy
Table Input Format and Size	String format with 2 table rows	62.0
	Markdown format with 2 table rows	68.8
	JSON format with 2 table rows	72.8
	JSON format with 5 table rows	75.2
Prompt Strategy	Roleplay prompt with JSON input	73.2
	Chain-of-thought prompt with JSON input	72.0
	Roleplay prompt with JSON + 5 table rows	71.2
Language Handling	Translate question to English + JSON with 5 rows	65.6
	Translate question, columns to English + JSON with 5 rows	58.4
	Spanish prompt + JSON with 5 rows	**76.4**

We use **DeepSeek-V3** [14] for all experiments instead of GPT-4o-mini, based on a baseline comparison using plain string input. DeepSeek-V3 achieved 62.0% accuracy, significantly outperforming **GPT-4o-mini** (smaller version of GPT-4o [15]), which reached only 48.4%. This justified our choice to use DeepSeek-V3 exclusively in the remaining evaluations.

We observe that providing input in JSON format consistently outperforms Markdown, with the highest performance under this group achieved when five rows are included (75.2%). This suggests that additional tabular context improves the model's ability to reason over structured data.

In terms of prompting strategies, roleplay-based prompting performs slightly better than chain-of-thought (73.2% vs. 72.0%), though both are competitive. Interestingly, combining roleplay with longer input (five rows) slightly lowers performance (71.2%), possibly due to input length affecting prompt clarity or token limit constraints.

Language handling plays a crucial role: using Spanish prompts yields the highest overall accuracy (76.4%), while translating only the question or both the question and columns results in a performance drop (65.6% and 58.4%, respectively). This indicates that translation can introduce semantic shifts or inconsistencies that degrade model understanding, especially when translating table headers.

Our experiment on development dataset in Table ... show that translation questions and columns from spanish to englih not help at all and drastically decrease the performance. Using markdown and json format for example rows show better performance than string example, As for test phase submission I submit three version first one is provide data rows in json format

4.2. Testing Phase

Table 2 presents the performance of the top five teams in the competition. Our team achieved the highest accuracy, tied with the `itunlp` team.

For the testing phase, we employed a more robust model DeepSeek-R1 [16], a reasoning model across several prompt configurations (detailed in Table 3). Our best performance was achieved using the raw, unprocessed dataset, reaching an accuracy of 87%. In contrast, preprocessing the dataset led to a slight drop in performance, yielding 85% accuracy.

Although translating prompts into Spanish showed promising results during the development phase, this approach resulted in a decrease in performance during testing, with accuracy dropping to 84%. These findings suggest that the model benefits more from maintaining the original data format and language fidelity in the final evaluation setting.

5. Conclusion

In this work, we investigated the impact of prompt design, input formatting, and language handling on table-based question answering using large language models. Our experiments, conducted using

Table 2
Top 5 teams in the competition.

Team	Accuracy (%)
itunlp	87
Our Team	**87**
hcerezo	85
Lys Group	78
quang3010	75

Table 3
Different Experimental Settings using DeepSeek-R1

Experiments Setting	Accuracy
raw data + english prompt + json format with 5 table rows	**87**
preprocessed data + english prompt + json format with 5 table rows	85
raw data + spanish prompt + json format with 5 table rows	84

DeepSeek-V3 and DeepSeek-R1, revealed that structured JSON inputs and longer table contexts (i.e., more rows) significantly improve model performance. Among prompting strategies, roleplay and chain-of-thought techniques offer moderate gains.

Notably, our results indicate that maintaining the original language (Spanish) in the prompt leads to better performance on the development set, but may generalize less effectively to the testing phase. Furthermore, we demonstrated that DeepSeek-V3 outperforms GPT-4o-mini on baseline tasks, justifying its use in all subsequent experiments.

Future work could explore dynamic prompting, multilingual fine-tuning, and hybrid symbolic-neural table reasoning to further enhance performance in low-resource or multilingual domains.

Acknowledgments

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

Declaration on Generative AI

During the preparation of this work, we used GPT-4 and Grammarly in order to: check grammer, spelling, and edit the content for clarity and coherence. After using these tools, we reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Osés-Grijalba, L. A. Ureña-López, E. M. Cámara, J. Camacho-Collados, Overview of PRESTA at IberLEF 2025: Question Answering Over Tabular Data In Spanish, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[2] J. Á. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[3] J. O. Grijalba, L. A. U. López, J. Camacho-Collados, E. M. Cámara, Towards quality benchmarking in question answering over tabular data in spanish, Proces. del Leng. Natural 73 (2024) 283–296. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6617.

[4] V. Zhong, C. Xiong, R. Socher, Seq2sql: Generating structured queries from natural language using reinforcement learning, CoRR abs/1709.00103 (2017).

[5] 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).

[6] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, W. Y. Wang, Tabfact: A large-scale dataset for table-based fact verification, arXiv preprint arXiv:1909.02164 (2019).

[7] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee, et al., Chain-of-table: Evolving tables in the reasoning chain for table understanding, arXiv preprint arXiv:2401.04398 (2024).

[8] D. Ji, L. Zhu, S. Gao, P. Xu, H. Lu, J. Ye, F. Zhao, Tree-of-table: Unleashing the power of llms for enhanced large-scale table understanding, arXiv preprint arXiv:2411.08516 (2024).

[9] P. Yu, G. Chen, J. Wang, Table-critic: A multi-agent framework for collaborative criticism and refinement in table reasoning, 2025. URL: https://arxiv.org/abs/2502.11799. arXiv:2502.11799.

[10] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[11] D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. K. Li, F. Luo, Y. Xiong, W. Liang, Deepseek-coder: When the large language model meets programming – the rise of code intelligence, 2024. URL: https://arxiv.org/abs/2401.14196. arXiv:2401.14196.

[12] 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. URL: https://arxiv.org/abs/2201.11903. arXiv:2201.11903.

[13] A. Kong, S. Zhao, H. Chen, Q. Li, Y. Qin, R. Sun, X. Zhou, E. Wang, X. Dong, Better zero-shot reasoning with role-play prompting, 2024. URL: https://arxiv.org/abs/2308.07702. arXiv:2308.07702.

[14] DeepSeek-AI, A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Guo, D. Yang, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Zhang, H. Ding, H. Xin, H. Gao, H. Li, H. Qu, J. L. Cai, J. Liang, J. Guo, J. Ni, J. Li, J. Wang, J. Chen, J. Chen, J. Yuan, J. Qiu, J. Li, J. Song, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Xu, L. Xia, L. Zhao, L. Wang, L. Zhang, M. Li, M. Wang, M. Zhang, M. Zhang, M. Tang, M. Li, N. Tian, P. Huang, P. Wang, P. Zhang, Q. Wang, Q. Zhu, Q. Chen, Q. Du, R. J. Chen, R. L. Jin, R. Ge, R. Zhang, R. Pan, R. Wang, R. Xu, R. Zhang, R. Chen, S. S. Li, S. Lu, S. Zhou, S. Chen, S. Wu, S. Ye, S. Ye, S. Ma, S. Wang, S. Zhou, S. Yu, S. Zhou, S. Pan, T. Wang, T. Yun, T. Pei, T. Sun, W. L. Xiao, W. Zeng, W. Zhao, W. An, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, X. Q. Li, X. Jin, X. Wang, X. Bi, X. Liu, X. Wang, X. Shen, X. Chen, X. Zhang, X. Chen, X. Nie, X. Sun, X. Wang, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yu, X. Song, X. Shan, X. Zhou, X. Yang, X. Li, X. Su, X. Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Y. Zhang, Y. Xu, Y. Xu, Y. Huang, Y. Li, Y. Zhao, Y. Sun, Y. Li, Y. Wang, Y. Yu, Y. Zheng, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Tang, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Wu, Y. Ou, Y. Zhu, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Zha, Y. Xiong, Y. Ma, Y. Yan, Y. Luo, Y. You, Y. Liu, Y. Zhou, Z. F. Wu, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Huang, Z. Zhang, Z. Xie, Z. Zhang, Z. Hao, Z. Gou, Z. Ma, Z. Yan, Z. Shao, Z. Xu, Z. Wu, Z. Zhang, Z. Li, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Gao, Z. Pan, Deepseek-v3 technical report, 2025. URL: https://arxiv.org/abs/2412.19437. arXiv:2412.19437.

[15] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen,

R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, Gpt-4 technical report, 2024. URL: https://arxiv.org/abs/2303.08774. arXiv:2303.08774.

[16] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, Z. Zhang, Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL: https://arxiv.org/abs/2501.12948. arXiv:2501.12948.