ChatGPT, be my teaching assistant! Automatic correction of SQL exercises

ChatGPT, be my teaching assistant! Automatic Correction of SQL Exercises

Luca Cagliero, Laura Farinetti, Jacopo Fior, Andrea Ignazio Manenti
DAUIN - Politecnico di Torino
Torino, Italy
(luca.cagliero, laura.farinetti, jacopo.fior)@polito.it
andreaignazio.manenti@studenti.polito.it

***Abstract*—The use of Large Language Models (LLMs) such as OpenAI ChatGPT to enhance teachers' and learners' experience has become established. The impressive capabilities of ChatGPT in solving Text2SQL problems prompts their use in database courses to solve SQL exercises. In this paper, we dig deep into ChatGPT abilities applied to SQL exercises. We quantitatively and qualitatively evaluate the performance of a *ChatGPT-as-a-SQL-assistant* on benchmark data, with particular attention paid to its ability to correctly detect syntactic and semantic errors, provide insightful judgment explanations, and assign grades comparable to those of human teachers. Furthermore, we also analyze the benefits of leveraging few-shot learning to adapt LLM responses to the expectation.**

***Index Terms*—Learning Analytics, SQL, Large Language Models, University-level education**

I. Introduction

The advent of Large Language Models (LLMs) such as OpenAI ChatGPT [1] has radically changed the prospects of usage of Natural Language Processing techniques in several application domains [2]. For instance, in the context of relational databases the use of deep learning to address the text-to-SQL task [3] has recently gained a significant interest. Given a relational database and a query on the database posed in natural language, Semantic Parsing (SP) is a well-known problem in the database community, whose goal is to reformulate the query as an SQL declaration that can be automatically executed by a database management system [4].

In this work, we explore the use of ChatGPT to support teachers and students in the context of an undergraduate database course, for SQL problem solving. The key idea is to leverage SP capabilities of a state-of-the-art LLM, i.e., OpenAI ChatGPT, to automatically examine and assess the solutions of SQL exercises provided by the course students. By performing a systematic, empirical comparison between human and GPT-provided corrections of SQL exercises, we highlight weaknesses and strengths of *ChatGPT-as-a-SQL-assistant*.

Previous attempts to use LLMs to enhance learning experiences in higher education have been made. For instance, learners can leverage ChatGPT to answer complex questions [5]–[7], to translate teaching content from one language to another [8], to write or explain programming codes [9], [10]. A preliminary attempt to use ChatGPT as a tutor to solve SQL assignments has recently been made in [11]. The authors have explored the capabilities of ChatGPT in a zero-shot setting, i.e., without considering any training example, to provide learners valuable assistance in generating SQL queries. In [12] the authors have also attempted to use a traditional sequence-to-sequence model, namely T5 [13], to automate grade SQL statements. Unlike traditional architectures like T5, pretrained LLMs such as GPT already incorporate larger amount of SQL- and database-related information thus not requiring extensive model fine-tuning.

In this work, we study to what extent ChatGPT can be adopted to correct SQL exercises, providing grades and educational feedback in an automated fashion. Unlike prior works, we aim to answer the following open Research Questions (RQ):

RQ1) Are the grades assigned by ChatGPT to SQL exercises comparable to those of human teachers?
RQ2) Is ChatGPT able to differentiate between syntactic and semantic errors?
RQ3) Is few-shot learning beneficial to improve the quality of ChatGPT evaluation compared to a zero-shot setting?

The remainder of this paper is organized as follows. Section II introduces the preliminary concepts and the benchmark dataset used in the experiments. Section III describes the *ChatGPT-as-a-SQL-assistant* methodology and the steps used to assess its performance on benchmark data. Section IV reports the outcomes of the quantitative and qualitative analysis carried out on the ChatGPT prompt-response pairs. Section V draws the conclusions of the present work and discusses future extensions.

II. Preliminaries

a) The Semantic Parsing (Text2SQL) Task: Given a free-text query $\mathtt{Q}$ and a relational database $\mathtt{DB}$, a Semantic Parser aims to map $\mathtt{Q}$ to a valid SQL statement $\mathtt{S}_Q$ that can be executed on $\mathtt{DB}$. Notice that each query can be correctly solved using multiple statements $\mathtt{S}_Q^1$, $\mathtt{S}_Q^2$, …, $\mathtt{S}_Q^n$.

We aim to assess the ability of the students of a relational database course to accomplish SP. Since Large Language Models are known to be effective in SP, we prompt the LLM with the query and submission details, asking the models to assess the correctness of the provided solution in place of the human teacher.

We explore two different scenarios, i.e., *Zero-Shot* and *Few-Shot Learning*. In a Zero-Shot Learning paradigm (hereafter denoted by ZSL for the sake of brevity) the LLM is prompted with just the strictly necessary information about the current submission under evaluation, i.e., a synthetic description of the database schema, the text of the query, the expected and student-generated solutions. Conversely, in a Few-Shot Learning Paradigm (FSL, in short) the prompt also contains few examples of other prompt-answers to exemplify the expected output structure and content. More details on the ZSL and FSL procedures are given in Section III.

b) The MovieDB benchmark: We carry out our analysis on the `MovieDB` relational dataset previously presented in [12]. It collects the 12,899 SQL statements submitted by the under-graduated students of a relational database course as well as the related inputs and outputs. More specifically, it contains

- The schema of each of the 16 database tables;
- The natural language text of 15 exercise queries on the database tables;
- The difficulty level of each query (5 difficulty levels are defined, from *L1* to *L5*, where *L1* is the easiest level and *L5* is the most difficult one);
- A draft solution for each exercise (i.e., the SQL declarations corresponding to each natural language query);
- The students' submissions for each exercise;
- The execution outcome of each submission (i.e., non-executable statement, partially correct answer, correct answer);
- For 15 random students' submissions for each of the 15 exercise queries: the grades assigned by 3 different teachers, from zero (worst) to 100 (best).

III. ChatGPT-as-a-SQL-Assistant

We selected the students' submissions graded by the teachers in the original `MovieDB` dataset and prompted ChatGPT to evaluate them. Then, we asked three other database course teachers to manually annotate the ChatGPT free-text outputs.

For each student submission ChatGPT returns both a grade and a free-text explanation of its evaluation. In the textual response, ChatGPT often also suggests how to correct or to improve the query and classifies the reported mistakes as *syntactic* or *semantic*. A key step in our research is to understand if the ChatGPT grades are correlated with the human ones or not and whether the explanations and suggestions provided by ChatGPT are correct or not.

A. Features

To answer the research questions introduced in Section I, for each submission we collected the values of the following features with the help of three database experts:

- *Mean teachers' grade*: The mean of the teacher grades in the original dataset.
- *ChatGPT grade*: The grade that is automatically assigned to the query by ChatGPT.
- *ChatGPT vs. humans grade difference*: For each query, the value is `Lower` if ChatGPT gave a lower grade with respect to the mean teachers' grade (being more "strict"), `Higher` if ChatGPT was more "generous".
- *ChatGPT explanation*: a free-text explanation of the evaluation generated automatically by ChatGPT.
- *ChatGPT syntax error*: A binary value assigned by the database expert by reading the ChatGPT explanation. It is `Yes` if the textual response by ChatGPT mentions the presence of syntax errors, `No` otherwise.
- *ChatGPT semantic error*: A binary value, assigned by the database expert by reading the ChatGPT explanation. It is `Yes` if the textual response by ChatGPT mentions the presence of semantic errors, `No` otherwise.
- *ChatGPT evaluation*: The binary class `Correct/Wrong` assigned by the database expert by reading the ChatGPT explanation. Notice that this class assignment relies only on the textual evaluation and is independent of the ChatGPT grade.
- *ChatGPT suggestion*: A categorization of the ChatGPT suggestion provided by the database expert into the following three classes: `Valuable`, `Not Valuable`, or `Missing`. For example, we deemed the ChatGPT suggestions as `Valuable` when ChatGPT says that the *DISTINCT* is not necessary, or it corrects the attributes of a *GROUP BY*; suggestions were classified as `Not valuable`, for instance, when it suggested to change the name of an alias or to have an alternative syntax for a *JOIN*. Finally, we assigned the `Missing` class when the ChatGPT explanation did not contain any suggestion.
- *True syntax errors*: A binary value (`Yes` or `No`) indicating whether the syntax errors reported in the explanation (if any) are actually present or not in the student's solution.
- *True semantic errors*: A binary value (`Yes` or `No`) indicating whether the semantic errors reported in the ChatGPT explanation (if any) are actually present or not in the student's solution.

B. Prompt definition

The prompt used to guide ChatGPT in the grading assignment is the result of a series of tests and optimizations. Specifically, we tested different prompts for both Zero-Shot and Few-Shot Learning. The prompts tested differ in the phrasing and in the structure of the request. For example, after some preliminary testing, we discovered that directly asking for a numeric rating and a short description of the errors is the best way to achieve consistency in the output structure. Similarly, the order in which we provide context to the model (e.g., the database schema, the query, etc.) was fine-tuned to obtain the most consistent and precise results over a small subset of the tested queries before the general testing.

Table I reports a couple representative examples of the selected ZSL and FSL prompts (see Column *Prompt*). For FSL we provided ChatGPT with additional context. Specifically, we included three to four different students' answers to the query and their corresponding ratings, selecting submissions representing the whole range of ratings (i.e., from 0 to 100). This version is more complex because it requires to grade some examples before starting to leverage the system. Also, considering the use of APIs, it would be more expensive given the higher number of required tokens.

IV. Experimental results

A. Correlation between grades

To address the Research Question RQ1, first of all we calculated the Pearson Correlation Coefficient (PCC) between the set of the average degrees assigned by the teacher and the set of the ChatGPT degrees. We calculated the coefficient considering (i) all the students' queries and (ii) the subsets of queries corresponding to exercises of the same difficulty level (from L1 to L5).

Figure 1 shows the coefficients. The overall coefficient shows a strong positive correlation between the two sets. Considering the difficulty level, the correlation coefficient for simple exercises (L1) is very high, but it tends to decrease as the difficulty level grows. This makes sense, because when exercises are difficult many different ways to make mistakes are possible: grading this kind of exercises is a complex task also for human teachers, and assigned scores may have sometimes a certain degree of subjectivity when teachers have to judge how severe an error is.

To support this statement, we analyzed the range of grades assigned by the three individual teachers to the same student query. The results are shown in Table II. The values in the table represent the average of the maximum difference among the grades assigned by the "human" teachers to the same query, calculated separately for each difficulty level, and overall. The third and the fourth columns consider separately the queries that could be defined as "correct" (average grade $\geqslant$ 85) and the queries that contain errors (the others). The data show that the variation increases as the difficulty level grows, and the variation is much higher when queries contain errors with respect to queries that are correct or almost correct.

We also calculated the correlation among the grades assigned by the three individual teachers: the average correlation coefficient is 0.938, much higher than the average correlation coefficient between each teacher and ChatGPT, which 0.642. This is not surprising, because teachers generally share an evaluation rubric.

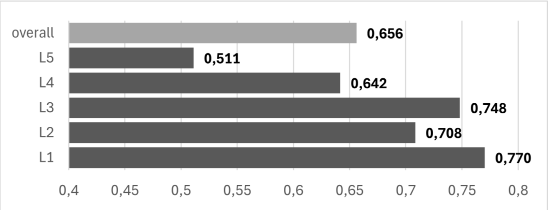

Fig. 1: Pearson Correlation Coefficients between the grades assigned by the teachers and the grades assigned by ChatGPT.

Figure 2 and Figure 3 visualize more in details the relationship between teachers' grades and ChatGPT grades by comparing the grades query by query.

Each point in Figure 2 is relative to one of the 224 analyzed students' query, and the shape of the point represents the difficulty level. The x-axis refers to the teachers' grades, while the y-axis refers to the ChatGPT grades. The line represents the ideal situation in which the grades are coincident; the points above the line represent queries for which ChatGPT was more "generous", the points below are the queries for which ChatGPT was "stricter". Figure 3 is a different representation, where the x-axis shows the query identifier and the two lines represents respectively the teachers' grade and ChatGPT grade. The queries are sorted by teachers' grade.

The two figures show that, even though the correlation between the grades is strong, degrees are not coincident for most of the queries, and that ChatGPT tends to be stricter. This is particularly evident when the teachers' grade is 100 (the maximum): only two of these queries got a 100 from ChatGPT.

The graph in Figure 4 gives another insight on this phenomenon: the x-axis shows the range of differences between ChatGPT grades and teachers' grades, and the y-axis reports how many queries belong to each interval. Results are shown separately per difficulty level, and positive values on the x-axis mean that ChatGPT was more generous, negative values mean the opposite. The graph confirms that ChatGPT tends to assign lower grades for all difficulty levels, but also that when the difficulty level grows the difference between the grades become much higher in both directions, confirming what has been reported in Figure 1.

B. Analysis of ChatGPT explanation text

To further address Research Question RQ1, besides the quantitative analysis on the scores, for each students' SQL query we performed a qualitative analysis on the corresponding ChatGPT generated text (ChatGPT explanation); this analysis ignored the score assigned by ChatGPT. After extracting manually the features *ChatGPT syntax error* and *ChatGPT semantic error*, we classified ChatGPT evaluation as `Correct`/`Wrong` (where `Correct` is a solution with none or very minor errors whereas `Wrong` is a solution with serious ones).

Table III shows the comparison of the teachers' evaluation (from the average degree assigned by the teachers) and ChatGPT evaluation. Considering the capability of ChatGPT to identify errors, we obtained a precision value of 0,48 and a recall value of 0,86. This result confirms that ChatGPT is quite good to find existing errors but tends to overestimate the seriousness of the students' mistakes, sometimes confusing actual mistakes with (for example) non optimized coding.

This is also supported by the analysis of the feature *ChatGPT suggestion*: Table IV shows that it has nothing to suggest for 3% of the queries only, and that in about half of the cases it does not provide any valuable suggestions.

TABLE I: Examples of ChatGPT prompts and responses.

Type	Prompt	Response
Zero-Shot Learning	Given this database schema: `(database schema)` Given the following question: `(Preamble of Exercise 3)` Considering that the following query would be the correct answer: `(Reference statement of exercise 3)` What grade from 0 to 100 would you give to the following query considering both semantics and syntax? `(submitted answer 3880)` Answer with a number followed by an explanation.	I would rate the given query a 50 out of 100. Explanation: Syntax: The syntax of the query is correct. It selects the id from the person table where the year born is equal to the maximum year born in the person table. The SQL syntax is clear and follows the standard structure for such queries. Semantics: The query selects the person(s) with the maximum year born in the person table. This would retrieve the oldest person(s) in the database, which is contrary to the requirements of the problem. The query does not address the requirement to find the second youngest person(s) in the database. The query lacks the logic necessary to identify the second youngest person(s) in the database based on the birth year. Overall, while the syntax of the query is correct, it fails to fulfill the requirements of the problem and does not achieve the intended outcome. Hence, it receives a low score of 50.
Few-Shot Learning	Given this database schema: `(database schema)` Given the following question: `(Preamble of Exercise 3)` The following submission corresponds to a rating of 100/100: `(submitted answer 852)` The following submission corresponds to a rating of 5/100: `(submitted answer 5169)` The following submission corresponds to a rating of 30/100: `(submitted answer 6692)` Considering that the following query would be the correct answer: `(Reference statement of exercise 3)` What grade from 0 to 100 would you give to the following query considering both semantics and syntax? `(submitted answer 3880)` Answer with a number followed by an explanation.	I would give the query a grade of 50/100. Explanation: The query correctly attempts to find the second youngest person in the database. However, it lacks the condition to ensure that the selected person is indeed the second youngest. It merely finds the person born in the same year as the youngest person. The query could benefit from incorporating the condition `(year_born < (SELECT MAX(year_born) FROM person))` to ensure it is specifically selecting the second youngest person. In summary, while the query correctly identifies the youngest person, it fails to properly identify the second youngest person due to the absence of the necessary condition.

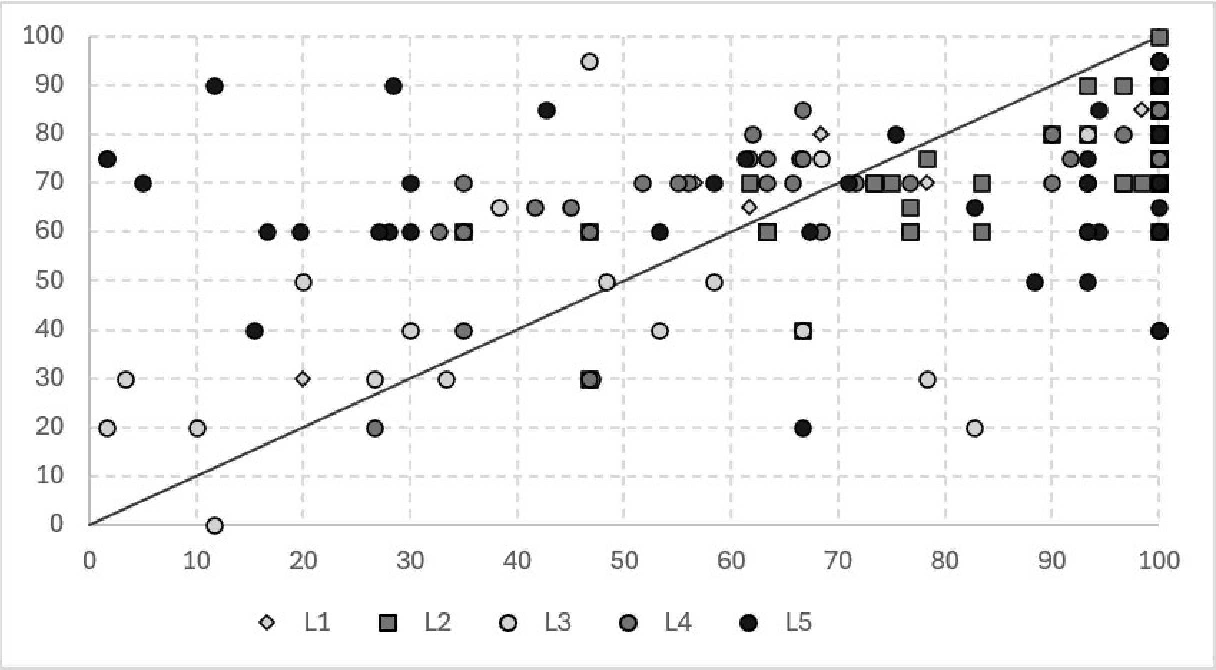

Fig. 2: Relationship between teachers' grades (on x-axis) and ChatGPT grades (on y-axis), query by query.

To address the Research Question RQ2, we evaluated ChatGPT capability to discriminate syntax and semantic errors by compiling the features *True syntax errors* and *True semantic errors*, that compare the syntactical/semantic errors reported in the ChatGPT explanation text with the actual presence of syntactical/semantic errors in the students' solution.

Table V shows the four combinations of the features' values, where:

1) *True Syntax Error & True Semantic Error* reports the queries for which ChatGPT discriminated correctly be-

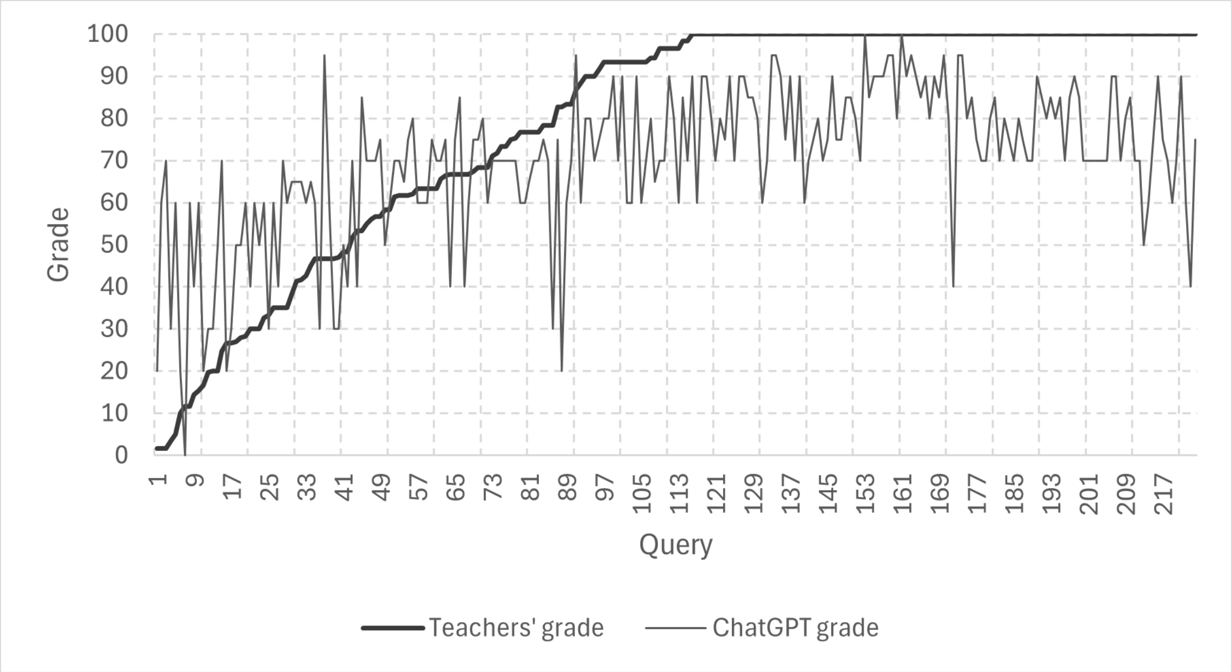

Fig. 3: Relationship between teachers' grades and ChatGPT grades, represented as lines. The x-axis is the query id, where queries are sorted by the teachers' grade.

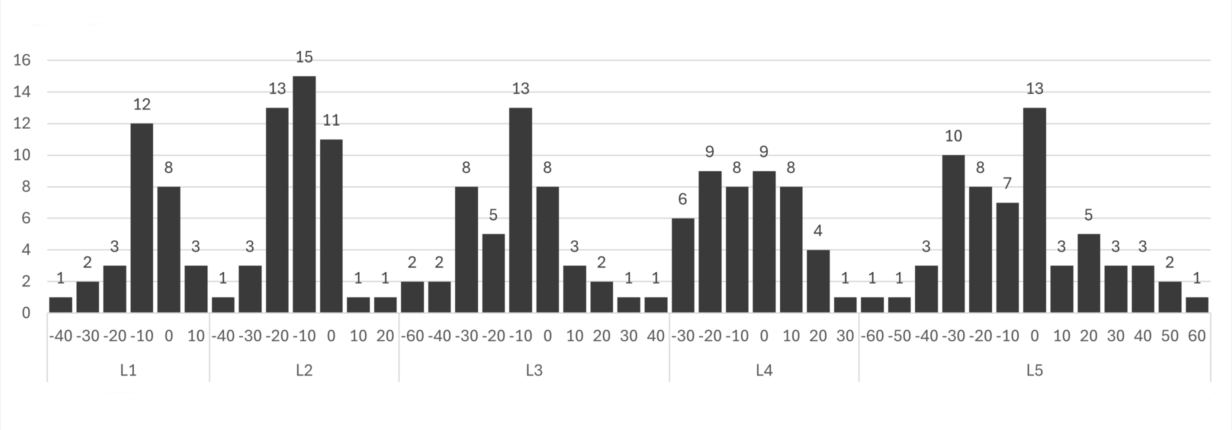

Fig. 4: Range of difference between teachers' grades and ChatGPT grades for each difficulty level.

TABLE II: Variation of the grades assigned by the three ”human” teachers to the same query.

Difficulty level	**Queries**		
	All	**Correct**	**Errors**
L1	2,76	0,21	15,00
L2	8,44	3,39	16,76
L2	7,58	0,74	17,83
L4	10,82	5,23	16,17
L5	10,95	5,79	17,26
Overall	**8,68**	**3,18**	**16,88**

TABLE III: Evaluation comparison, with number of students' queries classified in each group.

		ChatGPT eval.	
		Correct	**Wrong**
Teachers' eval.	**Correct**	51	83
	Wrong	13	77

TABLE IV: Value of ChatGPT generated suggestions.

Suggestion	**Query %**
Valuable	47,8%
Not valuable	49,1%
Missing	3,1%

tween syntax and semantic errors;

2) *True Syntax Error & False Semantic Error* represents the queries where ChatGPT reported an actual syntax error as a semantic one;

3) *False Syntax Error & True Semantic Error* represents the queries where ChatGPT reported an actual semantic error as a syntax one;
4) *False Syntax Error & False Semantic Error* represents the queries where ChatGPT was completely confused.

The values in the table represent the percentage of students' queries that belong to each category.

From this analysis we can see that ChatGPT is very good in finding existing syntax errors (Syntax True in Table V: 88,9%) but quite often it reports syntax errors as semantic ones (Semantics False in Table V: 38,4%).

TABLE V: ChatGPT capability to find and discriminate syntax and semantic errors.

	Query %
Syntax True & Semantics True	**56,3%**
Syntax True & Semantics False	**32,6%**
Syntax False & Semantics True	**5,4%**
Syntax False & Semantics False	**5,8%**

C. Zero-Shot vs. Few-Shot Learning

To address the Research Question RQ3 we compared the results achieved using ZSL with those obtained with FSL. Given the prompt-response examples in Table I, the ChatGPT explanations produced in ZSL show a better structure, with a clear distinction between syntactic and semantic aspects, and more insightful suggestions (e.g., *This would retrieve the oldest person(s) in the database, which is contrary to the requirements of the problem.*). Conversely, in the FSL setting ChatGPT tends to provide too synthetic and weakly informative explanations and gives overemphasis to the provided examples which, by construction, cannot be representative of all the possible failure cases.

Based on an extended comparative testing, the ZSL version performs on-par or better than its more elaborate counterpart. The main reason is that in the adopted ChatGPT version (i.e., ChatGPT 3.5 Turbo) the prompt fits up to 4,096 tokens, which are partly used to describe the database, formulate the query, and state the task. Therefore, it fits quite few training examples, yielding limited quality results improvements compared to ZSL. To overcome this limitations of LLM adaptation techniques, as future work we plan to (1) Extend *ChatGPT-as-a-SQL-Assistant* to support ChatGPT4, fitting up to 8,192 tokens, and (2) Fine-tune the LLM for the SQL evaluation task using lightweight fine-tuning strategies (e.g., [14]).

The explanations/suggestions provided by both ZSL and FSL tend to pay excessive attention to syntactic aspects such as the use of aliases (e.g., *the alias ”m” for the table ”movie” is used consistently throughout the query, which is good practice*), readability improvements (e.g., *The query uses the old-style join notation with commas in the ‘FROM‘ clause. While this is syntactically correct, it's considered outdated. Modern join syntax (using ‘JOIN‘ and ‘ON‘ clauses) is preferred for readability and maintainability)*, and optimization issues (e.g., *The query is not using the correct range for production years (1993, 1992, 1991)*). This challenges can be partly mitigated by providing more synthetic and informative training examples.

V. Conclusions and future work

The preliminary results of this on-going experiment show that ChatGPT is promising as a teaching assistant, being quite good in finding students' SQL errors, especially syntactical ones. It often confuses syntax and semantic errors, but with a more specific prompting maybe its discrimination capability could be improved.

About grades, ChatGPT is very strict but it has a personal scale of grades that often is proportional to the teachers' one. It tends to punish severely minor errors and imprecision, especially when the difficulty level of the exercise increases.

Automatic grading of SQL queries of course is not the goal, but if the capability of writing high-quality text explanations and to provide valuable suggestions is improved, ChatGPT could become a good companion for teachers (to help in the first step of the grading process) and even more for students, as a self evaluation tool. It is important to be very cautious in this direction though, because one of the main pitfalls of ChatGPT is overreliance: this could work only if students are able to approach critically ChatGPT output, and teachers can guide them in this task.

With this in mind, future research directions will explore the comparison between different LLMs and ChatGPT versions, the adoption of LLM fine-tuning, and the use of a Chain-of-Thought approach to incorporate reasoning steps on the most common logical flaws (e.g., applying the `IN` operator on the wrong attributes, excluding the wrong set of tuples).

References

[1] OpenAI, “Gpt-4 technical report,” *ArXiv*, vol. abs/2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[2] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, “Recent advances in natural language processing via large pre-trained language models: A survey,” *ACM Comput. Surv.*, vol. 56, no. 2, sep 2023. [Online]. Available: https://doi.org/10.1145/3605943

[3] G. Katsogiannis-Meimarakis and G. Koutrika, “A survey on deep learning approaches for text-to-sql,” *VLDB J.*, vol. 32, no. 4, pp. 905–936, 2023. [Online]. Available: https://doi.org/10.1007/s00778-022-00776-8

[4] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. R. Radev, “Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task,” *CoRR*, vol. abs/1809.08887, 2018. [Online]. Available: http://arxiv.org/abs/1809.08887

[5] K. Wang, J. Ramos, and R. Lawrence, “Chated: A chatbot leveraging chatgpt for an enhanced learning experience in higher education,” 2023.

[6] R. Sun and X. N. Deng, “Using chatgpt to enhance experiential learning of college students,” in *57th Hawaii International Conference on System Sciences, HICSS 2024, Hilton Hawaiian Village Waikiki Beach Resort, Hawaii, USA, January 3-6, 2024*, T. X. Bui, Ed. ScholarSpace, 2024, pp. 64–73. [Online]. Available: https://hdl.handle.net/10125/106384

[7] M. Wang, M. Wang, X. Xu, L. Yang, D. Cai, and M. Yin, “Unleashing chatgpt's power: A case study on optimizing information retrieval in flipped classrooms via prompt engineering,” *IEEE Transactions on Learning Technologies*, vol. 17, pp. 629–641, 2024.

[8] M. Javaid, A. Haleem, R. P. Singh, S. Khan, and I. H. Khan, “Unlocking the opportunities through chatgpt tool towards ameliorating the education system,” *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 3, no. 2, p. 100115, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772485923000327

[9] D. Popovici, "Chatgpt in the classroom. exploring its potential and limitations in a functional programming course," *CoRR*, vol. abs/2401.11166, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.11166

[10] E. Chen, R. Huang, H. Chen, Y. Tseng, and L. Li, "Gptutor: A chatgpt-powered programming tool for code explanation," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky - 24th International Conference, AIED 2023, Tokyo, Japan, July 3-7, 2023, Proceedings*, ser. Communications in Computer and Information Science, N. Wang, G. Rebolledo-Mendez, V. Dimitrova, N. Matsuda, and O. C. Santos, Eds., vol. 1831. Springer, 2023, pp. 321–327. [Online]. Available: https://doi.org/10.1007/978-3-031-36336-8_50

[11] N. Carr, F. R. Shawon, and H. M. Jamil, "An experiment on leveraging chatgpt for online teaching and assessment of database students," in *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, 2023, pp. 1–8.

[12] J. Wang, Y. Zhao, Z. Tang, and Z. Xing, "Combining dynamic and static analysis for automated grading sql statements," *J Netw Intell*, vol. 5, no. 4, pp. 179–190, 2020.

[13] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[14] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9