ChatGPT as a Solver and Grader of Programming Exams written in Spanish

Abstract

Evaluating the capabilities of Large Language Models (LLMs) to assist teachers and students in educational tasks is receiving increasing attention. In this paper, we assess ChatGPT's capacities to solve and grade real programming exams, from an accredited BSc degree in Computer Science, written in Spanish. Our findings suggest that this AI model is only effective for solving simple coding tasks. Its proficiency in tackling complex problems or evaluating solutions authored by others are far from effective. As part of this research, we also release a new corpus of programming tasks and the corresponding prompts for solving the problems or grading the solutions. This resource can be further exploited by other research teams.

1 Introduction

Large Language Models (LLMs) based on the Transformer architecture (Vaswani et al., 2017) have represented a paradigm shift in Natural Language Processing (NLP), making the previous state-of-the-art results and benchmarks obsolete. The release of ChatGPT by OpenAI in November 2022 meant a disruption in what was thought to be possible in generating human-like conversations (Forbes, 2022). This class of generative models has proved to be effective in a wide range of Natural Language Processing (NLP) tasks (Zhong et al., 2023; Mao et al., 2023).

These new tools have also demonstrated impressive capabilities for solving programming tasks. This is often attributed to the fact that their internal models have been exposed to a large number of programming examples during their training process (Zhong et al., 2023; Xu et al., 2022; Chen et al., 2021). LLMs can thus become a highly valuable asset to support different teaching activities in multiple computer-related courses and university degrees (Baidoo-Anu and Ansah, 2023). For example, we can exploit them to give support to students in problem solving, to suggest exercises and activities to the professors, or to assist in the grading processes. But this also comes with drawbacks, such as those related to plagiarism or cheating.

Some researchers have already tested ChatGPT for resolving programming problems, demonstrating human-level performance for simple tasks, but also finding that it struggles with complex data structures (Chang et al., 2023; Sarsa et al., 2022). However, most studies and benchmarks have been confined to exams written in English. Although the multilingual settings of many LLMs have made it possible to apply these models to other languages, the performance is often lower than that of English and more scientific efforts are needed to evaluate the benefits and limitations of these language models for other languages.

Furthermore, most previous studies focused on resolving simple programming tasks, such as basic coding exercises. Our evaluation addresses not only basic programming challenges, but also more complex exercises that require reasoning about computational complexity, decision making about algorithmic strategies, or selecting proper data structures (e.g., stacks or queues).

Another aspect that has received little attention is the role of LLMs as graders or assistants in evaluating the quality of solutions written by humans. By advancing our understanding on the grading abilities of AI agents, we can shed light on the feasibility of incorporating them into new (student-machine) learning activities or even exploiting them to automatically or semi-automatically grade academic assignments.

In this study, we evaluate ChatGPT's abilities to solve programming and algorithmic problems extracted from a real exam written in Spanish. The exam, which is the final test of a 1st-year/2nd-semester course on Programming within a BSc in Computer Science, covers a wide range of exercises, from basic coding exercises to more intricate reasoning tasks. We also assess here the AI's capacities to evaluate exams solved by students enrolled in the course from which the exam was taken.

Therefore, our contributions are:

- An evaluation of how well ChatGPT solves complex programming and algorithmic problems written in Spanish.
- A study of the feasibility of ChatGPT to act as an automatic evaluator for tests solved by university students.
- A detailed item-by-item analysis of the strengths and weaknesses of ChatGPT as a solver and grader of multiple programming exercises.
- A new corpus of programming tasks and the corresponding set of prompts to ask the models to solve problems or grade solutions. This new resource can be further exploited by other research teams to conduct further evaluations of LLMs for programming problem resolution. All the data and code of this research is freely available for the scientific community[1].

2 Method

2.1 ChatGPT as a Solver

We chose a real exam from a 1st year-2nd semester course on Programming, Linear Data Structures and Introduction to Computational Complexity. The exam was taken in May 2023 by 90 students from an accredited BSc degree in Computer Science. The average score of the students was 57.55% (std dev 20.29%), 26 of them did not pass (score below 50%), and 5 students scored above 90%.

It should be noted that this is an exam that tests not only basic coding skills, but also algorithmic and data structure concepts. The exam consisted of 7 questions (see Table 2, Appendix B), with varied types of expected responses, ranging from a full page to a short textual answer. Some of the questions involved the development of C code. In the original exam, two questions (#3 and #5) had two figures that further help to clarify the particular inquiry. Since ChatGPT[2] does not accept images, we opted for removing the images. In any case, the images were redundant (e.g. one represented the internal structure of the Abstract Data Type (ADT) list, whose code was given in the text of the question) and one can understand the question without having to see the image. However, we have to bear in mind that this may be a small disadvantage for the AI model. The evaluation of more advanced models, such as GPT-4 (OpenAI, 2023), was left for future work.

Each question was passed to ChatGPT through OpenAI's Python API. Two different prompt variants were tested: a simple one with almost no context (**Simple Prompt**) and a more sophisticated prompt including formatting instructions and system role (**Complex Prompt**), see Appendix A.1. The answers outputted by the AI model were given to the main instructor of the course (a professor in CS&AI), who assessed the model's solutions using the same criteria set for the official exam.

2.2 ChatGPT as a Grader

We also wanted to evaluate ChatGPT's capacity to assess the quality of human-made solutions. The official exams solved by the students were manuscript and, thus, we can hardly evaluate them all. Instead, we chose a sample of five exams, with a varied range of scores (94%, 74%, 66%, 50% and 38%), transcribed them and submitted them to the model's API. We sent each question individually and asked the model to provide a quality score (0%-100%), see Appendix A.2. Then, an overall grade was obtained by weighting the questions using the point scale established in the official exam.

3 Results

3.1 ChatGPT as a Solver

Table 1 shows the results achieved by ChatGPT in the exam. As can be seen in the first row, each question had a different number of points. To avoid any possible bias, the professor did not know which prompt generated each version of the responses.

The first noticeable result is that, for both variants, the model achieved a score above the required threshold to pass the final exam. This is not a minor outcome, since previous research has demonstrated that these models struggle with difficult data structure tasks (Chang et al., 2023). ChatGPT's grades are similar to those achieved by the average student. One might argue that matching human performance is profoundly meaningful. However, we see here two main sources of concern. First, the students examined are novice undergraduates in the first year

[1] https://anonymous.4open.science/r/chatgpt-grader-solver-exams-7BBF/

[2] We used gpt-3.5-turbo version for this experimentation

	Questions							Overall Grade
	1	2	3	4	5	6	7	
Points	1.5	1.5	1.5	1	1	2	1.5	
Simple Prompt	0%	83.3%	66.7%	100%	100%	100%	16.7%	**65%**
Complex Prompt	0%	43.3%	66.7%	100%	100%	75%	0%	**51.5%**

Table 1: Grades obtained by ChatGPT. The first row shows the maximum number of points per question

of training (most of them with only a few months of experience in programming). So, ChatGPT is not really matching expert-level performance. Second, ChatGPT's performance does not place it in a position to be utilised as an intelligent assistant. You can hardly exploit a tool that produces wrong results more than 30% of the time.

A second interesting result is that the use of a complex prompt did not help the model. The complex prompt was never better than the simple prompt. No single question got a better response from the complex prompt. The specification of system role, the "take-your-time" advise and the provided example do not seem to be useful and, perhaps, have introduced some confusion.

Regarding specific questions, both variants struggled with question number 1 (syntactic and semantic specification of an ADT) and number 7 (reasoning about an example of divide and conquer algorithm). In question 1, ChatGPT did not output a formal specification of the ADT, failed to provide a semantic description with proper algebraic notation and often resorted to not-allowed expressions (e.g. using integer expressions in C rather than generic numerical expressions). We conjecture that this might be related to the low availability of ADT examples with proper notations in the training data. Question #7 was about interpreting different levels of computational complexity of a divide and conquer solution, based on variables such as the number of subproblems, size of the subproblems and so forth. This type of conceptual question also made that ChatGPT failed loudly. For the rest of the questions (#2-#6) ChatGPT made a decent job. Three of them were mainly coding tasks (#3, #5, #6) and two of them (#2, #4) required some sort of reasoning but they refer to well-known computing examples (Fibonacci or list search). In some instances, the model did not follow the instructions and, rather than outputting solutions written in C, it provided correct solutions written in Python. Note also that ChatGPT did well on questions #3 and #5, which had a supporting image that the model could not see.

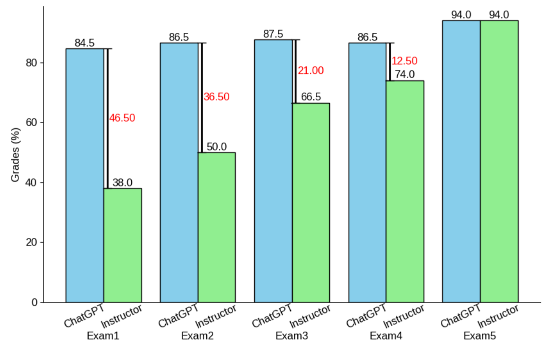

Figure 1: ChatGPT as a grader. For each exam solved by a student, the bars represent the score given by ChatGPT and the score given by the instructor of the course.

3.2 ChatGPT as a Grader

Figure 1 shows the grades assigned by ChatGPT and by the course's instructor to the five exams selected. The AI model clearly overestimates the quality of the solutions and, indeed, all exams got a high qualification (all of them above 84%). Even low quality solutions, such as the exam that officially got a 38% overall score, were assigned very good scores. This hardly positions ChatGPT as a tool to assist humans (professors or students) in the assessment of solutions for this type of exams.

Next, we analyse the individual question-by-question assessments, see Table 3 in Appendix C. We report the scores assigned and the deviation between ChatGPT and the instructor. The largest deviation was found in Question 1, on ADT specification. This result is in agreement with the findings in Section 3.1, in which the model also struggled to solve this exercise. Again, this suggests that the model has little knowledge about this topic or it is not able to transfer its knowledge to produce answers that comply with the instructions. The only decent grading by ChatGPT was done for question 4. This question was effectively solved by ChatGPT (see section 3.1) and, here, the model also shows reasonably good performance at evaluating question 4's answers written by students. These answers are short paragraphs explaining the computational complexity of a given search problem (traversing over a list). A somehow surprising result is that ChatGPT was highly effective at producing solutions for Question 5 but it drastically failed to assess the quality of Question 5 solutions written by students. This was a C function that implements a recursive process and ChatGPT was unable to effectively assess the quality of the functions written by students. Furthermore, ChatGPT's tendency to overrate the quality of the solutions was consistent over all types of questions[3].

Note also that ChatGPT assigned an overall score to the best exam that was the same score assigned by the instructor (94%). But this seems to be anecdotal, as the individual question-by-question scores (Exam5) show substantial deviations.

4 Discussion

The results of this study suggest that ChatGPT performs much better on solving exercises than it does on grading them. As a solver, it is worth noting the AI model's poor performance in some types of exercises (particularly in those that do not involve coding). In the near future, it will be beneficial to investigate the reasons for this poor performance. For instance, to understand whether it is related to the language (Spanish) in which the problems are expressed or due to a lack of training data for these types of exercises. The specific phrasing might have also played an important role, as ChatGPT performed poorly for exercise 7, while other questions –also about computational complexity– had much better answers from the model. This suggests that wording might have a strong influence on model's performance.

On the other hand, our results suggest that the model is useless to validate answers submitted by humans. Even with coding questions that the model solved very well, its assessment of solutions written by others was unsatisfactory.

5 Related Work

The development of Large Language Models (LLMs) has represented a paradigm shift for multiple NLP tasks (Brown et al., 2020; Kojima et al., 2022). In some cases, these models even reach human performance. For instance, previous studies evaluated GPT-4, demonstrating its ability for several reasoning-intensive tests, such as passing a technical entrance exam for a software engineering position (Bubeck et al., 2023).

The ability of these models to generate code has sparked interest in both the developer and teaching communities. In this direction, Chen et al. (Chen et al., 2021) introduced Codex, a model specifically fine-tuned for solving programming tasks. Other authors studied Codex's potential for solving Python tasks, demonstrating its effectiveness on the APPS benchmark (Hendrycks et al., 2021). The authors of (Sarsa et al., 2022) explored the possibility of integrating Codex in teaching duties, for example to generate coding exercises. Similarly, Xu and colleagues (Xu et al., 2022) conducted a systematic evaluation of different language models –both proprietary like Codex and open source– for coding completion and synthesis tasks. These authors also proposed a novel fine-tuned model that outperforms other alternatives for C programming. Chang et al. (Chang et al., 2023) stated that ChatGPT outperforms humans in simple coding assignments, but still struggles with data structure problems and graph theory.

In this paper, we have presented a novel approach to evaluate ChatGPT's programming abilities. We evaluated its capacities to solve assorted programming-related exercises, spanning multiple areas such as abstract data types, data structures and computational complexity. Another novelty that distinguishes our work from previous contributions is that we also studied the AI model's capacity as a grader. Furthermore, our study targeted the Spanish language, thus responding to the growing interest of the scientific community in evaluating the capabilities of LLMs in languages other than English (Deng et al., 2023).

6 Conclusion

In this study, we assessed ChatGPT's capacities to solve and grade programming exams (in Spanish) from an official university course. The results suggest that this AI model can only be used as a solver of basic coding exercises. Its abilities to solve and reason about intricate questions, and its capacity to assess solutions written by others are far from effective. The study of more sophisticated prompting strategies, such as those based on paraphrasing the original instructions, are left as future work.

[3]As a side note, we observe that most students would have loved to have ChatGPT as grader, replacing the official instructor.

Ethics Statement and Limitations

This research aims at evaluating the capabilities of the new generative AI models as a support tool in educational environments. Our study was constrained to exams written in Spanish because we were specifically interested in analysing how LLMs perform in languages other than English. Access to the exams was provided to us by the main instructor of the university course. The exams filled by the students were anonymised and their responses to the questions did not contain any personal or private reference.

The overall goal of this project is to gain an understanding on how the new AIs could help to automate or ease certain learning and grading tasks. This research does not pursue the elimination of human instructors from the university classes. As a matter of fact, we firmly believe that human-in-the-loop strategies are crucial to properly exploit the advantages and reduce the risks of AI-based agents.

We are also aware that more sophisticated models, such as GPT4, could perform better. But, currently, ChatGPT is a model that is freely available and it already has a huge user base worldwide (including many university students). Thus, our study was centered on the most popular and widely available platform. Anyway, in the future we will extend this research to other LLMs. We also recognise that more sophisticated prompt engineering could lead to better performance. We left this exploration for future research and we decided to employ here two initial types of prompts.

References

David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning. *Journal of AI*, 7(1):52–62.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023. Multilingual jailbreak challenges in large language models. *arXiv preprint arXiv:2310.06474*.

Forbes. 2022. Introducing chatgpt. [accessed April 4, 2023].

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. Gpteval: A survey on assessments of chatgpt and gpt-4. *arXiv preprint arXiv:2308.12488*.

OpenAI. 2023. Gpt-4 technical report. *arXiv:submit/4812508*.

Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*, pages 27–43.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. 2022. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–10.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.

Appendices

A Prompts

A.1 Prompts for Solving

A.1.1 Simple Prompt

Under this setting, no further instruction but the exam question was provided to the model:

Simple Prompt

User: *<QUESTION>*
System: ...

A.1.2 Complex Prompt

Under this setting, we specified a system role to give more context to the model about the task. Additionally, the large prompt specifies that the model can take "time to reason" (this is a recommended practice in these tasks) and, additionally, we also provided a demonstration, which consists of a natural language instruction (asking to build a hello world program) and the corresponding C code.

Complex Prompt

System Role: Estás respondiendo a las preguntas de un examen de informática centrado en el lenguaje de programación C.
User: Escribe un programa en C que escriba: Hello World
Assistant: El siguiente código está escrito en C:
include <stdio.h>
int main(void)
{
printf("Hello World");
return 0;
}
User: La siguiente es una pregunta de un examen de programación del primer año del grado de ingeniería informática. Hay bastante tiempo para responder, así que tómate el tiempo que sea necesario para dar una respuesta completa y razonada paso a paso. La pregunta está delimitada por < >. Además en el caso de que tengas que escribir código, primero especifica el lenguaje en el que está escrito, y luego escribe dicho código delimitándolo con ”' antes de la primera línea y después de la última, tal y como has hecho anteriormente:
<QUESTION>
System: ...

A.2 Prompt for Grading

Under this setting, the prompt asks the model to reason about the model's response to the question and, next, it asks the model to compare it against the provided response and, finally, give an overall quality score for the provided response.

m

Grading Prompt

User: Tu tarea es evaluar la respuesta a una pregunta de un examen de programación. Para ello razona primero tu respuesta y compárala con la respuesta proporcionada. No la evalúes hasta que no hayas respondido tú mismo a la pregunta. La pregunta está delimitada por <...> y la respuesta a evaluar está entre "...". El formato de tu respuesta debe ser el siguiente, respétalo sin añadir ningún comentario adicional y asegúrate de escribir una nota numérica sobre 100: Pregunta: (copia aquí la pregunta del examen entre <...>) Respuesta: (copia aquí la respuesta del alumno entre "...") Calificación: La nota es (nota sobre 100%)
<QUESTION>
"*RESPONSE*"
System: ...

B Exam Questions

Table 2 details the questions that made up the exam.

C Evaluating Results

Table 3 breaks down the grades assigned by both assessors.

Question	Topic	Response Type
#1	Syntactic and semantic formal specification of an abstract data type (ADT)	Full page
#2	Recursive implementation of Fibonacci. Reason about... i. worst/best/avg time complexity ii. type of algorithmic strategy iii. # invoked instances	3 short responses (1-2 sentences each)
#3	ADT List (internal data structure is given) i. & ii. implement functions get/create iii. & iv. worst case time complexity of functions get/create	2 short C functions + 2 one-word responses
#4	List. Worst/Best Time complexity of a given search problem	One paragraph
#5	Implementation in C of a recursive function that solves a given problem	Short C function
#6	Implementation in C of a function that uses an ADT Queue (with provided operations) to solve a given problem	C function
#7	Divide & Conquer Example. Reason about... i. parameters (# subproblems, split/aggregation costs, subproblem sizes) ii. compare two variants for the same problem iii. compare against 2 types of sequential search	3 short responses (2-3 sentences each)

Table 2: Exam Questions

	Question 1			Question 2			Question 3			Question 4		
	ChatGPT	Instructor	Dev	ChatGPT	Instructor	Dev	ChatGPT	Instructor	Dev	ChatGPT	Instructor	Dev
Exam1	100%	53.3%	+46.7%	100%	50%	+50%	50%	0%	+50%	90%	50%	+40%
Exam2	100%	16.7%	+83.3%	100%	33.3%	+66.7%	25%	66.7%	-41.7%	100%	100%	0%
Exam3	100%	50%	+50%	66.7%	83.3%	-16.7%	75%	60%	+15%	100%	100%	0%
Exam4	100%	16.7%	+73.3%	100%	83.3%	+16.7%	25%	33.3%	-8.8%	100%	100%	0%
Exam5	100%	100%	0%	93.3%	100%	-6.7%	66.7%	100%	-33.3%	100%	100%	0%
Avg.	100%	47.3%	50.7%	92%	70%	31.4%	48.3%	52%	29.8%	98%	90%	8%

	Question 5			Question 6			Question 7		
	ChatGPT	Instructor	Dev	ChatGPT	Instructor	Dev	ChatGPT	Instructor	Dev
Exam1	100%	0%	+100%	90%	50%	+40%	66.7%	50%	+16.7%
Exam2	100%	100%	0%	90%	50%	+40%	90%	16.7%	+73.3%
Exam3	90%	0%	+90%	80%	75%	+5%	100%	83.3%	+16.7%
Exam4	100%	100%	0%	100%	100%	0%	93.3%	93.3%	0%
Exam5	100%	100%	0%	100%	75%	+25%	100%	93.3%	+6.7%
Avg.	98%	60%	38%	92%	70%	22%	90%	67.3%	22.7%

Table 3: Comparison of scores assigned by ChatGPT and by the course's instructor.