

Davinci Goes to Bebras: A Study on the Problem Solving Ability of GPT-3

Carlo Bellettini¹^a, Michael Lodi^{1,2,3}^b, Violetta Lonati^{1,3}^c, Mattia Monga^{1,3}^d
and Anna Morpurgo^{1,3}^e

¹Università degli Studi di Milano, Milan, Italy

²Alma Mater Studiorum, Università di Bologna, Bologna, Italy

³Laboratorio Nazionale CINI 'Informatica e Scuola', Rome, Italy

Keywords: Bebras, GPT-3, Large Language Models, Computer Science Education.

Abstract: In this paper we study the problem-solving ability of the Large Language Model known as GPT-3 (codename DaVinci), by considering its performance in solving tasks proposed in the “Bebras International Challenge on Informatics and Computational Thinking”. In our experiment, GPT-3 was able to answer with a majority of correct answers about one third of the Bebras tasks we submitted to it. The linguistic fluency of GPT-3 is impressive and, at a first reading, its explanations sound coherent, on-topic and authoritative; however the answers it produced are in fact erratic and the explanations often questionable or plainly wrong. The tasks in which the system performs better are those that describe a procedure, asking to execute it on a specific instance of the problem. Tasks solvable with simple, one-step deductive reasoning are more likely to obtain better answers and explanations. Synthesis tasks, or tasks that require a more complex logical consistency get the most incorrect answers.


1 INTRODUCTION


Natural language processing, after decades of intense research, is now at a point in which it has the potential for impacting education heavily, since many cognitive tasks such as text summarization and translation to other natural languages or computer code can be effectively assisted by machine learning systems. Computer science educators should be prepared for this forthcoming revolution which will probably change many traditional learning objectives, such as code writing (Finnie-Ansley et al., 2022; Raman and Kumar, 2022).


Large Language Models (LLMs) are currently one of the most promising technology in this area; they are systems trained on a large corpus of human texts and are designed to perform text generation. LLMs have also been used to answer common sense and reasoning questions (Kojima et al., 2022; Clark et al., 2018).


In this paper we study the problem-solving ability of LLMs and consider their performance in solving tasks proposed in the “Bebras International Challenge on Informatics and Computational Thinking” (from now on, Bebras, <http://bebras.org>).


Bebras is a yearly contest organized in several countries since 2004 (Dagienè, 2010; Haberman et al., 2011), with almost three million participants worldwide. The contest, open to pupils of all school levels (from primary up to upper secondary), is based on tasks rooted on core informatics concepts, yet independent of specific previous knowledge such as for instance that acquired during curricular activities. Having in mind the goal of proposing an entertaining learning experience, tasks are moderately challenging and solvable in a relatively short time (three minutes on average). They are often based on multiple choice questions, but it is also common to have semi-open questions or even interactive ones. Bebras tasks were also used to measure improvements of students’ attitude to computational thinking (Straw et al., 2017) and they are used in many computer science educational activities beyond the contest (Bellettini et al., 2019; Chiazzese et al., 2018; Lonati et al., 2017; Dagienè and Sentance, 2016). There-

^a <https://orcid.org/0000-0001-8526-4790>

^b <https://orcid.org/0000-0002-3330-3089>

^c <https://orcid.org/0000-0002-4722-244X>

^d <https://orcid.org/0000-0003-4852-0067>

^e <https://orcid.org/0000-0003-0081-914X>

fore, Bebras tasks, with their focus on computer science and computational thinking education, provide a good benchmark to assess the potential of natural language processing in this context: can LLMs be used to answer Bebras tasks?

LLMs' approach to text generation is based on word prediction: they estimate the likelihood of a specific token (a word or even just a short sequence of characters) given a (very long) text prefix, and repeat this operation indefinitely to generate complex texts. The strategy has a long history, since it has its roots in Shannon's seminal work (Shannon, 1948), but in recent years it has reached a new level of sophistication and results.

One of the most popular LLM systems is GPT-3, developed by OpenAI (Brown et al., 2020). It is based on the GPT (Generative Pre-trained Transformer) architecture and it achieves strong performance on many benchmark datasets focused on translation, question-answering, and cloze tasks. An improved version of GPT-3 is also the basis of ChatGPT (<https://chat.openai.com>), a system fine-tuned for conversation that can be used to generate human-like responses. ChatGPT has demonstrated the capability of performing professional tasks such as preparing legal documents, and it is considered able to receive a B/B- grade in a Wharton MBA exam (Terwiesch, 2023). Although ChatGPT made the headlines for its remarkable performances, in this paper, we focus on GPT-3, that is probably less powerful, but better documented by scientific publications and available with an API, making it easier to do automated experiments. We used the version nicknamed `text-davinci-003` (from now on, DaVinci), trained on 570GB of Internet texts in order to tune $1.75 \cdot 10^{11}$ parameters (Brown et al., 2020), thus being one of the largest LLMs publicly available. DaVinci was able to correctly answer half of the challenging questions of a common sense reasoning dataset of multiple-choice questions collected from 3rd to 9th grade science exams, and two thirds of the easy questions (Brown et al., 2020). Thus we wanted to test how well DaVinci would perform at tasks designed to elicit problem solving and computational thinking abilities, such the ones proposed in the Bebras contest.

As the first impromptu attempts we did with DaVinci were rather impressive, we decided to test DaVinci performance with several Bebras tasks in order to study the following research questions:

RQ1: How often is DaVinci able to answer correctly on a collection of Bebras tasks?

RQ2: Are DaVinci's answers consistent among different runs?

RQ3: Does DaVinci perform better with some specific types of tasks?

RQ4: Are DaVinci's explanations sound and on-topic?

Problem solving is hard to learn and teach and LLMs could be useful with their ability to simplify texts, expand explanations, or even suggest recurring patterns. However, the effort required to produce reliable answers and the level of discerning competence needed by the user of generated texts is still not clear. We thus believe the answer to the proposed research questions is needed to understand the current state of the potential of LLMs in education.

Section 2 reviews some related works; Section 3 describes the design of the experiment we conducted; the results of the experiment are detailed in Section 4, and are then summarized and discussed in Section 5; finally Section 6 draws our conclusions.

2 RELATED WORK

In this section we review related work, mainly concerning the Bebras challenge and tasks (Section 2.1), and the use of LLMs to accomplish reasoning tasks (Section 2.2).

2.1 Bebras

The Bebras contest (Dagienė, 2010; Haberman et al., 2011), open to pupils from primary up to upper secondary schools, is based on tasks rooted on core informatics concepts and computational thinking, yet independent of specific previous knowledge such as for instance that acquired during curricular activities: Bebras tasks avoid the use of jargon and are especially aimed at a non-vocational audience, focusing on that part of informatics that should become familiar to everyone, not just computing professionals. The Bebras community (which includes delegates from more than 50 countries) yearly organizes an international workshop devoted to proposing a pool of tasks to be used by national organizers in order to set up the local contests. National organizers then translate and possibly adapt the tasks to their specific educational context.

Figures 1–5 show examples of Bebras tasks. The answer can be semi-open (as in the task of Figure 1), have a multiple choice (as for the tasks in the other figures) or even require a complex interaction with the contest platform (see for example (Bellettini et al., 2018)), but all the tasks are suitable for automatic correction. In most cases the text is complemented by some graphics, but in many cases the role of the pictures is only decorative. For this study, we considered

Beaver Xavier wants to represent some letters with binary digits 1 and 0. He notices that letters T and E are more frequent. He thus decides to give them a shorter representation and thus code the letters T, E, A, K, C, and R as follows:

T -> 1
 E -> 00
 A -> 0010
 K -> 0110
 C -> 1010
 R -> 1110

Xavier sent this coded message to Yvonne:
 1 0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0

Yvonne has already found that this messages ends with the letter E.

In letters, what is the complete message written by Xavier?

Figure 1: A semi-open Bebras task (Bebras id: 2021-CH-06, authored by the Swiss Bebras team. The correct answer is ‘TAKECARE’. In our classification (see Section 3.1) it was labelled as EX and AE.

only textual tasks, since a LLM is aimed at natural language processing activities.

2.2 LLMs and Reasoning Tasks

GPT-3 has been presented in (Brown et al., 2020) as a new autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model. Its performance was tested in the so called “few shot” setting, in which the model is given a few demonstrations of the task, but the neural network weights are not updated (as would happen instead in a “fine tuned” setting): a question typically has a context and a desired completion, and few-shot works by receiving 10–100 examples of context and completion, and then one final example of context, with the model expected to provide the completion. In particular, GPT-3 was tested on ARC (Clark et al., 2018), a common sense reasoning dataset of multiple-choice questions collected from 3rd to 9th grade science exams. On the ‘Challenge’ version of the dataset (filtered to questions which simple statistical or information retrieval methods are unable to answer correctly), GPT-3 is able to correctly answer 52% of the questions, 70% for the ‘Easy’ version.

(Kojima et al., 2022) shows that a careful choice of the prompt can greatly affect the performance. In particular, adding a “chain-of-thought” prompt (“Let’s think step by step:”) can improve GPT-3 performance on arithmetic word problems from 18% to 79%, or even 93% with the additional help of “few shots”. However, the performance on com-

mon sense and other reasoning tasks datasets remains much lower (52–68%) and rather independent of the “chain-of-thought” prompt.

(Terwiesch, 2023) tested ChatGPT, an improved version of GPT-3 (GPT-3.5) “fine-tuned” to conversational tasks (*i.e.*, the model was specifically trained on conversational examples), on five questions of the Operations Management course within the MBA program at Wharton. ChatGPT did very well at basic operations management and process analysis questions including those based on case studies, providing correct answers and excellent explanations. However, it made surprising mistakes in relatively simple calculations at the level of 6th grade math. Moreover, ChatGPT was not capable of handling more advanced process analysis questions, even when they were based on fairly standard templates, in particular with multiple products and problems with stochastic effects.

(Saparov and He, 2022) studied the performance of GPT-3 with respect to logical (deductive) reasoning and they found it capable of making correct *individual* steps, but when multiple valid deduction steps are available, GPT-3 is not able to systematically explore the different options.

OpenAI also offers a version of GPT-3 fine tuned on publicly available code from GitHub, known as Codex (Chen et al., 2021). Codex has been found rather effective on typical introductory programming problems, scoring within the top quartile of CS1 students (Finnie-Ansley et al., 2022). However, it failed on problems which placed restrictions on what features could be used in the solutions. (Sarsa et al., 2022) found that Codex can also be very useful in helping create programming exercises that are sensible, novel and applicable, although the explanations provided might contain several inaccuracies.

(Raman and Kumar, 2022) starts from the assumption that LLMs will soon have a great impact on education, and especially computer science education. Thus they suggest to reduce the emphasis on code writing which characterizes current CS programs, since this skill will become less useful when LLMs will be able to write sensible code.

3 EXPERIMENT DESIGN

In this section we illustrate how we selected the tasks to submit to DaVinci, the queries we asked, and the analysis methods we used.

3.1 Selection of Tasks

We submitted to DaVinci 54 tasks, written in English. All tasks were selected from the pools of tasks prepared during the International Bebras Task Workshop 2022 and 2021 (317 tasks overall).

As DaVinci is a language model, we needed tasks that do not rely on graphical information. We looked for:

- textual tasks, *i.e.*, tasks in which all information needed to solve them is expressed in the text, without referring to charts or interactive elements. Images, if present, have a decorative role only;
- quasi-textual tasks, *i.e.*, tasks that contain an image/diagram whose informative content can be easily put into textual format (for example, in task 2022-CA-03 a simple textual description of the food present in each plate was used to replace the pictures of the plates).

When we selected a task, we also tagged it according to the cognitive task it requires to be solved, using macro levels inspired by the revised Bloom taxonomy (Anderson and Krathwohl, 2001):

- *Execution* (EX): a typical EX task describes a procedure, and solvers are asked to execute it on a specific instance of a problem;
- *Analysis/Evaluation* (AE): a typical AE task requires to analyze a situation or a procedure and draw some conclusion, *e.g.*, establishing what is the best solution to a problem, given certain constraints;
- *Synthesis/Creation* (SC): a typical SC task asks to invent a solution or sometimes a program, given some general specifications or an interpreter.

Among the 54 selected tasks, 18 of them were classified as EX, 42 as AE, 17 as SC (note that each task could be categorized with more than one level). In our pool there are fewer SC and EX tasks because they often require a formalization with some graphical notation/block language; adapting them for a textual model would have been a too-deep distortion of the original tasks. The summary of the classification is shown in Table 1.

Moreover, we tagged with MC the Multiple Choice tasks, which ask for a choice among a short list of options, to distinguish them from the ones with a semi-open question; 38 tasks were tagged as MC.

3.2 Queries

We used the OpenAI API to query DaVinci. Since a query does not give a deterministic answer, we repeated each query 10 times with the same parameters.

Table 1: Classification of the tasks

Labels	# MC tasks	# semi-open tasks
AE	16	5
EX	7	2
SC	2	0
EX, SC	1	0
AE, SC	6	7
AE, EX	6	1
AE, EX, SC	0	1
<i>Total</i>	<i>38</i>	<i>16</i>

Each query consisted in fact of two API calls with two prompts. The first prompt was:

Q: <Bebras task>

A: Let's think step by step.

Note that «Q:» and «A:» are part of the prompt. The introduction of «Let's think step by step» follows (Kojima et al., 2022) and it stimulates a chain-of-thought completion, in order to get an “explanation” of the answer, not just an answer. Then the second prompt asks to produce exactly one answer in a given form, for example «an arabic numeral» or «a letter among A through D» (for MC tasks), or «a string of six alphabetic characters»:

Q: <Bebras task>

A: Let's think step by step.

<previous step by step explanation>

Therefore the answer (<form of the answer>) is

The repetition of the first prompt and answer is needed, because the system has no memory of previous calls.

A parameter of the model called *temperature* regulates the “creativity” of the generated answer (the higher the temperature, the more verbose and varied the answer). We tried with values of temperature of 0.0 (almost a deterministic system), 0.1, 0.7 (the value known to be used in ChatGPT), and 0.9. In total we did 2160 queries, 40 per each of the 54 tasks, ten for each temperature.

3.3 Analysis Methods

To answer to RQ1, RQ2, and RQ3 we analyzed only the final answer provided by DaVinci (*e.g.*, the answer to the second prompt) and did not consider the explanation obtained with the first prompt («Let's think step by step»), which instead was instrumental in stimulating a chain-of-thought completion and improve the performance (Kojima et al., 2022). We automatically checked the correctness of DaVinci's answers over the 40 queries for each of the 54 tasks. In particular we counted the number of tasks such that

all answers are correct, the majority of answers is correct, there is at least one correct answer, all answers are wrong.

For RQ2, we also counted how many different answers each task received. As for RQ3, we considered the format of questions (multiple choice MC or semi-open), and the cognitive classification (EX, AE, SC), as defined in Section 3.1). In order to assess their impact on answers' correctness, we fitted the following logistic regression model (1) which posits that the probability p_i of answering correctly to task i is a function of the linear expression $\alpha + \beta_{EX}EX_i + \beta_{AE}AE_i + \beta_{SC}SC_i + \beta_{MC}MC_i$, where α is a parameter used to take into account unknown factors, EX_i, AE_i, SC_i, MC_i are 1 if i is marked respectively as EX, AE, SC, MC, 0 otherwise, and $\beta_{EX}, \beta_{AE}, \beta_{SC}, \beta_{MC}$ are the parameters that measure the impact of the marking and that we want to estimate, giving to the model uninformative prior distributions (see (Gelman et al., 2020) for further details on this approach).

$$\begin{aligned}
 \alpha &\sim Normal(0,5) & (1) \\
 \beta_{EX} &\sim Normal(0,5) & \beta_{AE} \sim Normal(0,5) \\
 \beta_{SC} &\sim Normal(0,5) & \beta_{MC} \sim Normal(0,5) \\
 p_i &= \text{logit}^{-1}(\alpha + \beta_{EX}EX_i + \beta_{AE}AE_i \\
 &\quad + \beta_{SC}SC_i + \beta_{MC}MC_i) \\
 y_i &\sim Bernoulli(p_i)
 \end{aligned}$$

For RQ4, we did not perform a systematic analysis of the explanations provided by DaVinci, which would require full manual inspection and qualitative analysis. However we looked at a sample of them and started collecting some observations about issues occurring with DaVinci's explanations.

4 RESULTS

In this section we present the results of our experiment: first, in Section 4.1 we consider the answers obtained by DaVinci, then in Section 4.2) we focus on the explanations.

4.1 DaVinci's Answers

RQ1. *How often is DaVinci able to answer correctly on a collection of Bebras tasks?* The results of the queries are summarized in Table 2. Overall DaVinci was correct 612 times over 2160 queries (28.3%); considering only the queries at temperature 0.0 (the most deterministic ones), DaVinci was correct 165 times over 540 queries (30.6%). The tasks

with a majority of correct answers are 16 (29.6%). There are 35 tasks (64.8%) with at least one correct answer (out of 40 tries), the average number of correct answers being 17.5 (standard deviation: 14.9). Conversely, the answers were all wrong for 19 tasks (35.2%).

RQ2. *Are DaVinci's answers consistent among different runs?* Most tasks received different answers on the various queries; Table 3 shows the variability of the answers, which increases with the temperature. Sometimes this variability is useful to find a correct solution, but in general it makes the answer more uncertain: overall only 2 tasks received 40 correct answers, whereas, with temperature 0.0, 15 tasks always received the correct answer (out of 10 tries). The number of different answers was high even for multiple-choice tasks. In fact, the number of variants can be even higher than the proposed choices, since DaVinci sometimes puts together more options (e.g., 'A and B', even if the answer was supposed to be only one among A, B, C, and D).

The number of different answers increases with the semi-open questions, e.g., for task 2021-CH-06 presented in Section 2.1 (see Figure 1). In this case, the answer obtained by DaVinci was always wrong; overall, the answers given were 21 (in order of decreasing frequency: TEAKCR, AKTECR, AKTERE, TEACKR, AKTCER, TEACKER, EKCAE, TEAKCER, TACKER, TEAKCRTE, TEAKCRTEAKCRE, TEKCRE, TEKACRE, TAKER, TEKAER, AKRETEE, TEAKCRE, AKRECEKE, TAKCEER, TEACER, EAKCRE); In 28 cases over 40, the explanation does not even match the hint given in the text ("Yvonne has already found that this messages ends with the letter E").

RQ3. *Does DaVinci perform better with some specific types of tasks?* The results concerning the cognitive classification of tasks (EX, AE, SC, see Section 3.1) are summarized in Table 4: the probability to get a correct answer increases when a task is marked as EX, and decreases when is marked with SC; the impact of an AE marking is less clear. The table contains also the result for the tasks marked as MC, for which the probability to get a correct answer increases as well.

One of the task that was solved correctly in all trials is 2022-CZ-05 (see Figure 2); the task mostly relies on linguistic skills, as it could be solved by simply associating 'people live' with 'population'.

Another task with a high rate (80%) of correct answers is task 2022-PK-01 (see Figure 3). The task

Table 2: Correct answers by temperature, ten queries per temperature, 40 in total for each of the 54 tasks, divided in 38 MC and 16 semi-open.

	Overall		temperature			
	MC	semi-open	0.0	0.1	0.7	0.9
<i>Always correct</i>	2	0	15	11	2	3
<i>Majority correct</i>	12	4	16	17	11	12
<i>At least one correct</i>	29	6	20	22	31	30

Table 3: Number of different answers per task (irrelevant typographical differences such as spaces, punctuation, spelling were not considered). The number of options in MC tasks is normally 4, but there is one task with 8 and one with 5.

	all tasks	Overall		temperature			
		MC	semi-open	0.0	0.1	0.7	0.9
<i>mean</i>	5.5	3.2	11.0	1.4	2.1	3.4	3.5
<i>std. dev.</i>	5.7	1.3	8.0	0.9	1.4	2.3	2.4
<i>max.</i>	26	7	26	6	6	10	10

Table 4: Posterior distributions of logistic regression parameters.

	mean	95% high density interval	
		from	to
α	-1.1	-1.5	-0.8
β_{EX}	0.7	0.4	0.9
β_{AE}	-0.1	-0.3	0.2
β_{SC}	-0.7	-0.9	-0.5
β_{MC}	0.2	0.0	0.5

Pavel has an access to a database of the United Nations. This database contains the following data about each country:

- Area [km²]
- Population [2015]
- Density [1/km²]
- Life expectancy

Pavel wants to compare how many more people live in other country than his.

What data Pavel needs to use to find this out?

- A) population
- B) life expectancy and area
- C) density and life expectancy
- D) life expectancy

Figure 2: Task 2022-CZ-05, authored by the Czech Republic Bebras team. The correct answer is ‘A) population’. In our classification (see Section 3.1) it was labelled as AE.

was marked as EX, and the correct answer can be obtained with a simple, one-step deductive reasoning.

4.2 DaVinci’s Explanations

RQ4. *Are DaVinci’s explanations sound and on-topic?* When reading DaVinci’s explanations, we rec-

Puffy has a pile of six clean shirts on her bed. She has to put them into her four drawers. She takes one shirt at a time from the top of the pile, and puts into a drawer. She starts with the top drawer, then uses the second from the top, and so on. When she has put a shirt into the bottom drawer, she starts again from the top.

Into which drawer will she put the last shirt? (Note that the top drawer is labeled with 1, the second from the top with 2, the third from the top with 3 and the last one with 4)

- A) 1
- B) 2
- C) 3
- D) 4

Figure 3: Task 2022-PK-01, authored by the Pakistan Bebras team. The correct answer is ‘B) 2’. In our classification (see Section 3.1) it was labelled as EX.

ognized some recurring features. We will summarize and illustrate them in the next paragraphs, by using excerpts concerning the Bebras tasks presented in Figures 1–5. In order to appreciate the presentation, before continuing readers are invited to read the tasks and try to solve them on their own.

There Are Cases Where Both the Answer and the Explanations Are Flawless.

Task 2022-CZ-05 (see Figure 2) was solved correctly in all trials. In at least one case, the explanation produced was indeed flawless:

Pavel wants to compare how many more people live in other countries than his. To do this, he needs to compare the population of his country with the population of other countries.

Beavers Ben and Tom exchange messages consisting of 12 digits 0 and 1. Because Beaver Bonnie understands their messages they decided to encode them. In the first encoding step they replace a pair of consecutive digits by a character A, B, C or D:

00	A
01	B
10	C
11	D

In the second step they replace a pair of consecutive characters using the system below:

AA	0
AB	1
AC	2
AD	3
BA	4
BB	5
BC	6
BD	7
CA	8
CS	9
CC	A
CD	B
DA	C
DS	D
DC	E
DD	F

yielding the final encoded message.

If the final message is C13, what was the initial message in digits 0 and 1?

- A) 110000010011
- B) 101010101111
- C) 101010111011
- D) 101011001001

Figure 4: A Bebras task that DaVinci solves correctly in all the 40 queries (Bebras id: 2021-RS-04, authored by the Serbian team). The correct answer is 'A) 110000010011'. In our classification (see Section 3.1) it was labelled as EX, AE.

Therefore, the data Pavel needs to use is population. The answer is A) population.

Another task with a high rate (80%) of correct answers and consistent explanations is task 2022-PK-01 (see Figure 3). With temperature 0.0 and 0.1, DaVinci always produced the same correct answer and the same rigorous explanation (with minor variations):

She will put the first shirt into drawer 1, the second shirt into drawer 2, the third shirt into drawer 3, the fourth shirt into drawer 4, the fifth shirt into drawer 1, and the sixth shirt into

drawer 2. Therefore, the last shirt will be put into drawer 2. The answer is B) 2.

Contradictions and False Statements.

The examples above are more an exception than a rule. Even when the answer is correct, in many cases the explanations are incomplete, contain errors, contradictions, or false statements.

Charithra wants a red, electric car. It needs to be able to drive at least 150km without needing to be recharged.

Shopper
 Colour: Red
 Range: 120km
 Fuel: Electric

Comet
 Colour: Black
 Range: 250km
 Fuel: Electric

AG12
 Colour: Red
 Range: 200km
 Fuel: Electric

IT Car
 Colour: Red
 Range: 350km
 Fuel: Petrol

Which car should Charithra get?

- A) Shopper
- B) Comet
- C) AG12
- D) IT Car

Figure 5: Task 2022-UK-06, authored by the British team, classified by us as AE, see Section 3.1. The correct answer is 'AG12'.

For instance, let us consider task 2022-UK-06 (see Figure 5), where DaVinci often correctly identified the AG12 car as the solution. The following explanation (obtained at temperature 0.1), does not explained why the IT Car is excluded (which is however correct, as the IT Car, also a red car, is not electric as required):

Charithra wants a red car, so the Shopper and AG12 are both good options. However, Charithra needs a car that can drive at least 150km without needing to be recharged, so the AG12 is the best option as it has a range of 200km. Therefore, Charithra should get the AG12.

Another explanation for the same task (temperature 0.1) introduces false statements and novel constraints, to justify the (correct) answer:

Charithra wants a red, electric car that can drive at least 150km without needing to be recharged. The Shopper does not meet the range requirement, so it is not the right choice. *The Comet* and AG12 both *meet the colour and fuel requirements*, but *the AG12 has a higher range of 200km, so it is the best choice for Charithra*. The IT Car does not meet the fuel requirement, so it is not the right choice. Therefore, the answer is C) AG12.

The emphasis on the false statements (it is not true that the Comet meets the color requirement, it is not true that the AG12 has a higher range) and the new constraint (Charithra does not need the highest range).

For task 2022-PK-01 (see Figure 3), sometimes DaVinci gave a wrong answer, and the explanations contain inconsistencies (either self contradictions, or contradictions w.r.t. the task's text, or both), e.g.:

The last two shirts will go into drawer 4 and then *drawer 1 again*. So, the answer is D) 4.

She will first put a shirt in the top drawer (1), then the second from the top (2), then the third from the top (3), and *finally the last one (4)*. So, the last shirt will be put in Drawer 1. The answer is A) 1.

It is worth mentioning that in all the examples above, despite the explanations are incomplete or contain errors, they propose in fact a reasonable approach to tackle the tasks.

There Are Cases with Correct Answer and Plainly Wrong Explanation.

Task 2021-RS-04 (see Figure 4) is one of the two that DaVinci always got right. The explanations given, however, are plainly wrong. For example (temperature 0.0):

The final message is C13, so the second step must have been CD and AB. This means that the first step must have been 11 and 01. Therefore, the initial message in digits 0 and 1 was 110000010011.

Or even (temperature 0.9):

First of all, we know in the second step that C13 corresponds to AA00. Then, in the first step we know that AA would correspond to 00 and 00 would correspond to A. Therefore, 110000010011 was the initial message in digits 0 and 1.

The 100% of right answers seems to be an artifact of the options given. This was confirmed by querying DaVinci with a semi-open version of the task, asking for a string of 12 digits 0 and 1: in this case the answer was *never* the expected one. The alternatives were: 110011001100 (20 times), 101100000000, 110011000000, 001001000000, 1100110000, 110101110000, 110101000000, 11010010, 1100DDAA, 110011011100, 1100101100, 001100000000, 101100001111, 1001110000, 100100000000, 100100000000, 10001011, 0011010000, A011B001F; 8 times out of 40 the 12 digits requirement is not satisfied. These results were obtained in post-hoc queries with the same experimental setting described in Section 3.2 and were not counted in Tables 2, 3, and 4.

Explanations Sound Coherent and On-Topic.

Overall DaVinci's linguistic ability (syntax, glossary, rhetoric) is impressive; at a first reading, the explanations sound coherent, on-topic, authoritative. As seen above, however, this apparent coherence of explanations does not go along with their logical accuracy, consistency, and rigor.

Here is an example of an apparently sound explanation, used to justify a *wrong* answer for task 2022-UK-06 (see Figure 5):

Charithra wants a red car, so the Shopper and AG12 are both good options. However, Charithra also wants an electric car, so the IT Car is not an option. Finally, Charithra needs a car that can drive at least 150km without needing to be recharged, so the Comet is the best option. Therefore, Charithra should get the Comet.

The explanation is contradictory since the Comet car is considered after it was not even selected in previous statements.

5 SUMMARY OF RESULTS AND DISCUSSION

Our study is able to give a first answer to the research questions we posed in Section 1, as summarized in what follows.

RQ1. In our experiment, DaVinci was able to answer with a majority of correct answers about one third of the Bebras tasks we submitted to it (16 out of 54). This is not much, especially when one considers that 36 tasks require a choice between 4 options, so even a randomized solver would get the

right answer approximately one fourth of the time. This performance is poor, but it actually confirms previous results: both (Kojima et al., 2022) and (Brown et al., 2020) report similar figures for “common sense”, “reasoning” and “multistep arithmetic” tasks. Even (Terwiesch, 2023) reports problems with simple mathematical operations.

RQ2. DaVinci answers are also rather erratic, and even when they are not, there is no guarantee that they are correct. On average one gets 5.5 different answers to the same question, 3.2 even considering only multiple choices.

RQ3. DaVinci seems to be better suited for tasks which describe a procedure, asking to execute it on a specific instance of the problem. Synthesis tasks, instead, are harder. Tasks which require a simple, one-step deductive reasoning, an ability exhibited by LLMs (Saparov and He, 2022), are more likely to obtain better answers and explanations.

RQ4. DaVinci linguistic fluency is remarkable, and it is too easy to be fooled by such a competent conversationalist. In general, the text generated by an LLM often appears to be coherent and on-topic. Sometimes the explanations are indeed remarkably good, in other cases they are questionable or plainly wrong. A deeper scrutiny is needed to recognize inconsistencies, errors, gaps in the chain of thoughts, that occur frequently. In fact, DaVinci produces plenty of connectives (so, therefore, but, however, we can assume, this means, since . . . must, etc.), and the reader must pay close attention to discern when they are justified and when they contribute to make an argument look sounder than it is.

6 CONCLUSIONS

As we showed, the apparent coherence of DaVinci’s answers and explanations does not go along with their logical accuracy, consistency, and rigor. The perceived logical soundness of the explanations is in fact only in the “eye of the beholder” (Bender et al., 2021). Humans tend to attribute intentions and beliefs to any interactive entity, even when *by construction* the interlocutor does not have any. Indeed human-human communication is a jointly constructed activity and we use the same facilities for producing language that is intended for audiences not co-present with us (readers, listeners, watchers at a distance in time or space) and in interpreting such language when we encounter

it (Bender et al., 2021). However, DaVinci (and in fact all LLMs) is nothing more than a “stochastic parrot” which somewhat repeats the patterns seen during its training, and yet it is almost impossible to avoid to see in its answers a straightforward reasoning process.

DaVinci’s authoritative style can be deceptive and it requires a developed critical thinking attitude to debunk. On the one hand, this is worrisome because it adds to the already imperative need for critical sense and ability to research and verify sources imposed by the modern digital society, in which there is plenty of information made public to a broad audience by people with malicious intent or simply not experts in that field. On the other hand, this very need can be exploited to implement more active and constructive teaching activities, in which students are required precisely to develop such critical sense by challenging themselves in analyzing and correcting the seemingly sensible answers of an LLM.

All in all, we believe DaVinci and its underlying GPT-3 system is an outstanding achievement for natural language processing, with human-like fluency and a surprising ability to generate on-topic sentences. But this should not be confused with *understanding*, even in its limited sense of framing an object of knowledge in order to support consistent inferences. The generated sentences do not need to comply with any coherent model, although often they in fact seemingly do. From an educational viewpoint this seems to be the greatest danger. Any user of such systems should be fully aware of its intrinsic limitations, even if its surface features are, by design, aimed at blurring the border between what can be done automatically and what instead requires actual ingenuity.

An almost gone generation of engineers sometimes still complains about the replacement of slide rules with pocket calculators: the former tool required an expert guess *before* the calculation, thus improving the number sense of its users. Stochastic parrots may be useful to understand better when a problem does not, in fact, require full fledged reasoning with a consistent chain of thought. Or, their generated content can be a challenging training ground for discovering rhetorical explanations that do not stand up to an analytical screening. We should expect that LLMs will improve, maybe mixing the stochastic part with some kind of model-based inference. But currently the hype is bigger than the actual performance: despite the impression of obsolescence that traditional education methods give when compared to this new technology, the benefits of introducing it in mainstream practice are all to be proven.

We can also say something about Bebras tasks, since they appeared to hard to grok by a LLM: they

still require a cognitive effort that is still not easily automated. In other words, notwithstanding their toy size, they are a useful educational playground to practice problem solving and computational thinking.

ACKNOWLEDGMENTS

M. Lodi's work has been supported by the Spoke 1 "FutureHPC & BigData" of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di "campioni nazionali di R&S (M4C2-19)" — Next Generation EU (NGEU).

REFERENCES

- Anderson, L. W. and Krathwohl, D. R., editors (2001). *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. Longman, New York, complete edition.
- Belletini, C., Carimati, F., Lonati, V., Macoratti, R., Malchiodi, D., Monga, M., and Morpurgo, A. (2018). A platform for the Italian Bebras. In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018) — Volume 1*, pages 350–357. SCITEPRESS.
- Belletini, C., Lonati, V., Monga, M., Morpurgo, A., and Palazzolo, M. (2019). Situated learning with Bebras tasklets. In S., P. and V., D., editors, *Informatics in Schools. Fundamentals of Computer Science and Software Engineering. ISSEP 2019.*, volume 11913 of *LNCS*, pages 225–239. Springer, Cham.
- Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. (2021). Evaluating large language models trained on code. Technical Report ARXIV.2107.03374, arXiv.
- Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., and Tosto, C. (2018). Exploring the effect of a robotics laboratory on computational thinking skills in primary school children using the bebras tasks. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM'18*, pages 25–30, New York, NY, USA. ACM.
- Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. Technical Report ARXIV.1803.05457, arXiv.
- Dagienė, V. (2010). Sustaining informatics education by contests. In *Proceedings of ISSEP 2010*, volume 5941 of *Lecture Notes in Computer Science*, pages 1–12, Zurich, Switzerland. Springer.
- Dagienė, V. and Sentance, S. (2016). It's Computational Thinking! Bebras Tasks in the curriculum. In *Proceedings of ISSEP 2016*, volume 9973 of *Lecture Notes in Computer Science*, pages 28–39, Cham. Springer.
- Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., and Prather, J. (2022). The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. In *Australasian Computing Education Conference, ACE '22*, page 10–19, New York, NY, USA. Association for Computing Machinery.
- Gelman, A., Hill, J., and Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press.
- Haberman, B., Cohen, A., and Dagienė, V. (2011). The beaver contest: Attracting youngsters to study computing. In *Proceedings of ITiCSE 2011*, pages 378–378, Darmstadt, Germany. ACM.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. Technical Report ARXIV.2205.11916, arXiv.
- Lonati, V., Monga, M., Morpurgo, A., Malchiodi, D., and Calcagni, A. (2017). Promoting computational thinking skills: would you use this Bebras task? In Dagienė, V. and Hellas, A., editors, *Informatics in Schools: Focus on Learning Programming: Proceeding of the 10th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2017*, volume 10696 of *Lecture Notes in Computer Science*, pages 102–113, Cham. Springer International Publishing.

- Raman, A. and Kumar, V. (2022). Programming pedagogy and assessment in the era of AI/ML: A position paper. In *Proceedings of the 15th Annual ACM India Compute Conference, COMPUTE '22*, page 29–34, New York, NY, USA. Association for Computing Machinery.
- Saparov, A. and He, H. (2022). Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. Technical Report ARXIV.2210.01240, arXiv.
- Sarsa, S., Denny, P., Hellas, A., and Leinonen, J. (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, ICER '22*, page 27–43, New York, NY, USA. Association for Computing Machinery.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.
- Straw, S., Bamford, S., and Styles, B. (2017). Randomised controlled trial and process evaluation of code clubs. Technical Report CODE01, National Foundation for Educational Research. Available at: <https://www.nfer.ac.uk/publications/CODE01>.
- Terwiesch, C. (2023). Would Chat GPT get a Wharton MBA? A prediction based on its performance in the operations management course. <https://mackinstitute.wharton.upenn.edu/2023/would-chat-gpt3-get-a-wharton-mba-new-white-paper-by-christian-terwiesch/>.

