Efficiency Divide: Comparative Analysis of Human & Neural Network Algorithm Development

Petr Gallus, Tomáš Ráčil and Tomáš Šlajs
University of Defence, Brno, Czech Republic

petr.gallus@unob.cz
tomas.racil@unob.cz
tomas.slajs@unob.cz

Abstract: The paper delves into a comparative analysis between human and artificial intelligence (AI) capabilities in algorithm development, with a specific focus on the challenges presented in the "Advent of Code." The research thoroughly investigates the performance of Generative Pre-trained Transformers (GPTs), such as ChatGPT and Bard, in solving intricate algorithmic problems and benchmarks these results against those achieved by human participants. A sizeable portion of the study is dedicated to understanding the nuances of prompt engineering in AI and how it affects the problem-solving process, alongside exploring the choice of programming languages used by both AI and humans. The methodology of the research is extensive, involving the participation of both AI models and human subjects, who vary in their levels of programming expertise. This approach allows for a comprehensive evaluation of the correctness and efficiency of solutions, along with the time taken to resolve the given problems. The results from this study reveal intriguing insights. While AI models like GPTs demonstrate an impressive speed in problem resolution, they often fall short in accuracy when compared to human problem-solvers, particularly in tasks demanding deeper contextual understanding and creative reasoning. Furthermore, the study delves into the impact of time constraints on the effectiveness of problem-solving strategies employed by both AI and humans. It finds that under strict time constraints, AI models can quickly generate solutions, but these solutions may lack the depth and accuracy found in those devised by human participants. This aspect of the research highlights the trade-off between speed and precision in AI-driven problem solving. The research extends its implications beyond mere performance comparison. It suggests the potential for a synergistic approach where the computational efficiency and rapid problem-solving abilities of AI can be effectively combined with the nuanced understanding and creative problem-solving skills inherent in humans. This hybrid approach could redefine the future landscape of programming and algorithm development. The study not only provides a critical analysis of the capabilities of AI in the realm of algorithmic problem-solving but also paves the way for future exploration into the collaborative dynamics of human-AI interaction in programming. It highlights the evolving role of AI in programming and underscores the importance of balancing AI's computational prowess with human creativity and adaptability in solving complex, real-world problems.

Keywords: Human Intelligence, Artificial Intelligence, Algorithm Development, Large Language Models, Comparative Analysis, Problem-Solving Efficiency

1. Introduction

In the rapidly evolving landscape of software development, the intersection of human ingenuity and artificial intelligence (AI) is creating new frontiers in algorithm development. This paper explores this intersection by delving into the "Advent of Code," a unique platform that presents a series of coding challenges known for their complexity and requirement for innovative approaches. The "Advent of Code," created by Eric Wastl, stands out due to its intricately designed puzzles that often demand creative problem-solving strategies. The difficulty of these challenges is highlighted by the fact that in the latest competition, out of 231,994 participants, only 3,047 were able to complete the final algorithmic task [1]. This not only underscores the complexity of the tasks but also sets a rigorous benchmark for assessing coding proficiency.

The advent of Generative Pre-trained Transformers (GPTs) like ChatGPT and Bard has ushered in a new era in programming. These AI models are not only revolutionizing the landscape of code generation but are also expanding their reach across various domains. Their potential to assist in software development raises pivotal questions about the future role of human programmers. This paper aims to investigate whether these advanced neural networks can tackle complex algorithmic problems, akin to those found in the "Advent of Code," and what implications this may have for the programming community.

An integral aspect of this exploration is the role of prompts in directing the functionality of GPTs. Traditionally, the efficacy of a GPT's response heavily relies on the quality of the prompt provided. This study seeks to understand whether intricate prompt engineering is essential for solving complex algorithmic tasks or if GPTs can effectively operate with minimal guidance. This has significant implications for the usability of LLMs (Large Language Models) by individuals without programming expertise.

Furthermore, the choice of programming language is a crucial factor in the realm of software development. In the current scenario, where the emphasis is on delivering efficient and rapid solutions, the selection of a programming language is often dictated by these goals. This research aims to juxtapose the algorithmic solutions provided by university students on the "Advent of Code" platform with those generated by the latest LLMs. This comparison seeks to evaluate the potential of LLMs to solve complex tasks autonomously, which could signify a change in basic assumptions in the role of programmers, reducing it to system architects or requirements specifiers.[2]

This study is not just an academic exercise; it is a foray into the potential future trajectories of programming as a profession. By examining the capabilities of LLMs in contrast to human problem-solving skills in a challenging environment like the "Advent of Code," we aim to uncover insights that could shape the future direction of software development and the role of human programmers in it.

2. Methodology

This study embarked on a comprehensive examination of algorithmic problem-solving skills, juxtaposing human capabilities against advanced neural networks. The primary platform for this investigation was the "Advent of Code" 2023 challenges, a well-known series of programming puzzles that escalate in complexity.

2.1 Selection of Challenges and Participants

The research was bifurcated into two distinct phases. Initially, a broad assessment was conducted on the first six tasks of the "Advent of Code" which is a digital event during the Christmas season each year, where participants are presented with daily coding challenges suitable for various skill levels and can be tackled using any programming language. This inclusive initiative is widely utilized for diverse purposes including interview preparation, company training, university coursework, and as a competitive speed contest.[2]

These six tasks were followed by an extended evaluation up to day 15, ceasing when human participants were unable to progress further. A total of 23 participants, ranging from first-semester undergraduates to doctoral candidates with 1 to 10 years of programming experience, were enlisted. Additionally, 27 first-year students participated in a time-bound challenge, attempting to solve as many tasks as possible from the first six within a 90-minute time limit.

2.2 Neural Networks Usage

The neural networks employed were Bard, ChatGPT 3.5, ChatGPT 4.0 and custom GPT. For ChatGPT 4.0 and custom GPT, complete tasks from the "Advent of Code" website were inputted along with the respective data. Due to input length limitations with ChatGPT 3.5 and Bard, only essential parts of the tasks were extracted for generating Python code, which was then tested against the provided data. Each LLM had up to three attempts to generate a correct solution before moving to the next task.

2.3 Custom configured GPT

The "Own GPTs" feature of ChatGPT represents a groundbreaking advancement in customizable AI technology. It allows users to create and tailor their custom GPT models to cater to specific needs and tasks. A notable example of this innovation is "Adventer", a custom GPT developed specifically to tackle coding challenges from the renowned Advent of Code event. Adventer [Figure 1] is designed to excel in understanding and solving complex programming problems, making it an invaluable tool for programmers participating in these challenges. By analysing code structures, algorithms, and logic, Adventer assists in deciphering and solving intricate puzzles, highlighting the versatility and adaptability of custom GPTs in specialized fields.[4]

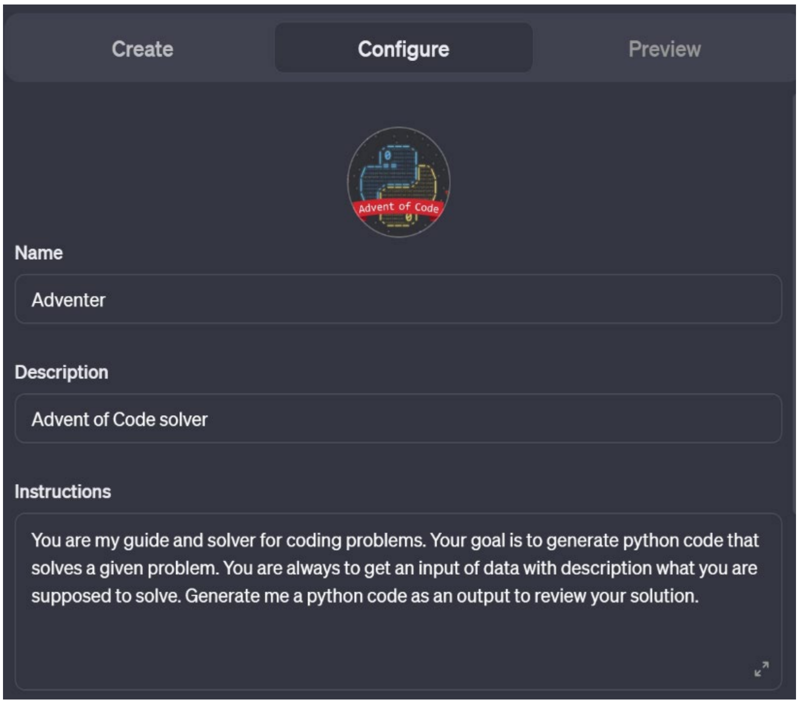

Figure 1: Custom GPTs default configuration. Source: [own]

2.4 Prompt Importance

Prompts play a pivotal role in the functionality of Generative Pre-trained Transformers (GPTs). They serve as the primary interface through which users communicate their requests or tasks to these language models. Prompts are input sentences or questions that guide the GPT to generate specific outputs. The effectiveness of a prompt influences the quality of the GPT's response. This is because the language model relies on the context provided by the prompt to generate relevant and accurate information.[5]

In the realm of coding, particularly when solving coding challenges with GPT models, the role of prompts becomes even more crucial. A well-crafted prompt can effectively direct the GPT to analyse a coding problem, provide solutions, or even write and debug code snippets. The importance of prompts in this context lies in their ability to transform abstract coding challenges into specific tasks that the GPT can understand and execute. The prompt not only provides context but also sets the scope and limitations within which the GPT operates, making it an indispensable tool in leveraging GPTs for coding challenges.[6]

2.5 Algorithmization and Programming Languages

Algorithms are fundamental to the field of computer science and play a crucial role in solving coding challenges. An algorithm is a set of well-defined instructions or rules designed to perform a specific task or solve a problem. In programming, algorithms are implemented using a programming language, which serves as a tool to express these instructions in a form that a computer can execute. The choice of algorithm and programming language significantly impacts the efficiency and effectiveness of the solution to a coding challenge. For example, pseudocode is often used in teaching problem-solving skills in programming, emphasizing the development of an algorithmic solution before coding it in a specific language.[7]

2.6 Comparison Criteria and Data Collection

The primary focus of comparison has been the correctness of solutions and the time taken to solve the problems. Notably, when the LLMs provided correct solutions, they do so at a speed unattainable by human programmers. Data collection for human participants involves two approaches: extracting data from a private leaderboard on the "Advent of Code" site and using a custom-built testing website. The website enables more precise tracking of the number of attempts and the time taken to solve each task.

2.7 Ethical Considerations

Ethical integrity was maintained throughout the study. Participant data were anonymized, and all individuals were informed about the purpose of their participation and the use of collected data.

2.8 Limitations and Assumptions

The study acknowledges several limitations. For the first group of participants, uninterrupted problem-solving was not feasible due to the time-consuming nature of the tasks and their academic obligations. The second phase limited students to a 90-minute window time limit, which, while sufficient for initial tasks, was restrictive for later challenges. The potential for more extended testing was curtailed by logistical constraints and the availability of participants.

2.9 Reproducibility

To ensure the reproducibility of our experiments, we meticulously detail our methodology, emphasizing the configurations of various neural network models such as ChatGPT 3.5, ChatGPT 4.0, Bard, and our tailored GPT setup. Our experimental design includes selecting coding challenges from the "Advent of Code" and engaging participants across a spectrum of programming expertise. While we strive for precision in the technical setup, we acknowledge the inherent variability in human performance due to differences in education, experience, and regional influences, which may affect the development of relevant skills.

Our comprehensive data collection strategy, utilizing both a private leaderboard and a custom testing platform, was designed to capture a wide array of data points, including solution correctness and problem-solving speed. Ethical integrity was maintained throughout the study, with a focus on participant anonymity and transparent data usage. This approach ensures that our findings are both reliable and replicable, notwithstanding the natural variance in human participant abilities.

Replicating this study requires adherence to our detailed setup and acknowledgment of the human element's variability. This variability underscores the complexity of drawing comparisons between human and AI problem-solving capabilities. By providing clear guidelines on our experimental configuration and participant selection, we aim to enable other researchers to validate our results and further explore the dynamic interaction between humans and AI in solving complex challenges, while recognizing the unique factors influencing human performance.

3. Results

In the context of the 'Advent of Code' challenges, the study provides a nuanced understanding of the performance of human participants compared to AI models, including ChatGPT 4.0, 3.5, and Bard.

3.1 Correctness Analysis of Task Performance

3.1.1 LLMs (Large Language Models) Performance

The LLMs demonstrated a consistent level of correctness, averaging between 16.67 % and 33.33 %. Its performance shows a degree of stability across the tasks, albeit with a lower correctness rate. This consistency might indicate the models' uniform approach to problem-solving, regardless of the task complexity.

3.1.2 Human Participants – Limited Time Limit

This group displayed a significant drop in correctness after the initial tasks, with correctness rates of 9.26 % and 3.70 % for the first two tasks and 0 % for subsequent tasks. The data suggests a struggle in coping with the complexity or specific requirements of the later challenges within the limited time limit.

3.1.3 Human Participants – Unlimited Time Limit

Exhibiting higher correctness rates, this group achieved 88.64 % and 86.36 % in the first two tasks. The rates gradually decreased in the subsequent tasks but remained notably higher than the other groups. This trend highlights their ability to adapt and apply effective problem-solving strategies, especially when not constrained by time.

3.1.4 Task-wise Correctness Comparison

The plot comparing correctness rates (Figure X) across tasks illustrates these trends vividly. LLMs maintain a flat line of moderate success, while the unlimited time limit group shows a descending curve, indicating higher success in initial tasks. The limited time limit group's curve sharply declines after the first two tasks.

3.1.5 Implications of Correctness Analysis

The correctness analysis underscores the distinct strengths and limitations of each group. While LLMs offer a consistent but limited accuracy, human participants, particularly with an unlimited time limit, demonstrate a higher degree of adaptability and accuracy, especially in tasks in which understanding context and applying creative solutions are crucial.

These findings suggest the potential of combining the computational efficiency of LLMs with the creative and adaptive problem-solving skills of humans, particularly in scenarios where accuracy and understanding of complex problems are vital.

Table 1: LLMs and Human Participants Correctness Analysis. Source: [own]

Group	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
LLMs	16.67% (std: 0.41)	33.33% (std: 0.52)	0.0%	33.33% (std: 0.52)	16.67% (std: 0.41)	16.67% (std: 0.41)
Human Participants – Limited time limit	9.26% (std: 0.29)	3.70% (std: 0.19)	0.0%	0.0%	0.0%	0.0%
Human Participants – Unlimited time limit	88.64% (std: 0.32)	86.36% (std: 0.35)	63.64% (std: 0.49)	75.0% (std: 0.44)	54.55% (std: 0.50)	52.27% (std: 0.51)

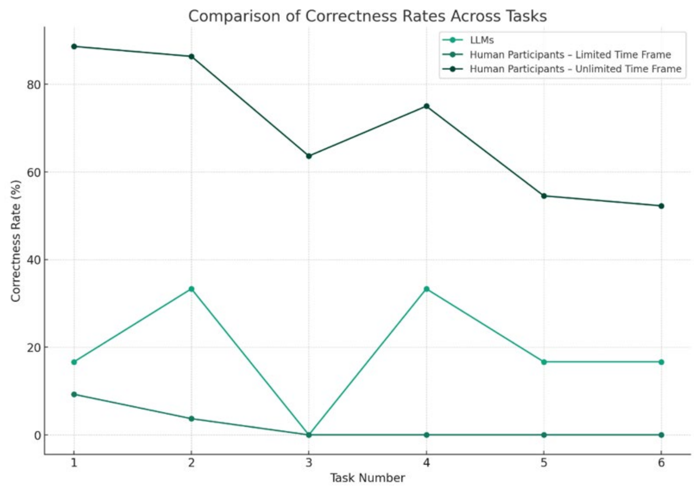

Figure 2: LLMs and human participants Correctness Analysis Graph. Source: [own]

The plot above [*Figure 2*] visually compare the correctness rates of the three groups – LLMs (Large Language Models), Human Participants with a Limited time limit, and Human Participants with an Unlimited time limit – across the first six tasks of the 'Advent of Code' challenges.

Key Observations:

- LLMs show a consistent correctness rate, averaging around 16.67% to 33.33%, with some variability in tasks 2 and 4.
- Human Participants with a Limited Time Limit exhibit lower correctness rates, particularly in tasks beyond the first two, where the rate drops to 0%.

- Human Participants with an Unlimited Time Limit demonstrate higher correctness rates, especially in the initial tasks, with a gradual decrease in later tasks but still maintaining high rates of correctness. [*Table 1*]

3.2 Time Efficiency Analysis

The time spent on tasks reveals striking differences between AI models and human participants:

3.2.1 LLMs

Exhibited consistent and rapid task completion, with an average time ranging from 34 to 544 seconds for various tasks. The standard deviation, where applicable, was low, indicating a uniform approach across tasks. [*Table 2*]

3.2.2 Human Participants – Limited Time Limit

Showed considerable investment in time, with an average of about 18.74 minutes for Task 1 and 15.16 minutes for Task 2. The high standard deviation suggests a wide variation in time spent among participants. [*Table 2*]

3.2.3 Human Participants – Unlimited Time Limit

This group had a much broader range of time spent, with averages from 14 to 85 hours for different tasks. The notably high standard deviation indicates a diverse set of strategies and time investments among these participants. [*Table 2*]

Table 2: LLMs and Human Participants Time Spent Analysis Table. Source: [own]

Group	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
LLMs	34 sec	38 sec (std: 4.24 sec)	-	52.5 seconds (std: 7.78 sec)	544 sec	53 sec
Human Participants – Limited time limit	1124.47 sec (~18.74 min	909.66 sec (~15.16 min)	-	-	-	-
Human Participants – Unlimited time limit	50874.85 sec (~14.13 h)	229329.71 sec (~63.70 h)	118959.93 sec (~33.04 h)	308699.52 sec (~85.75 h)	148960.75 sec (~41.38 h)	164948.74 sec (~45.82 h)

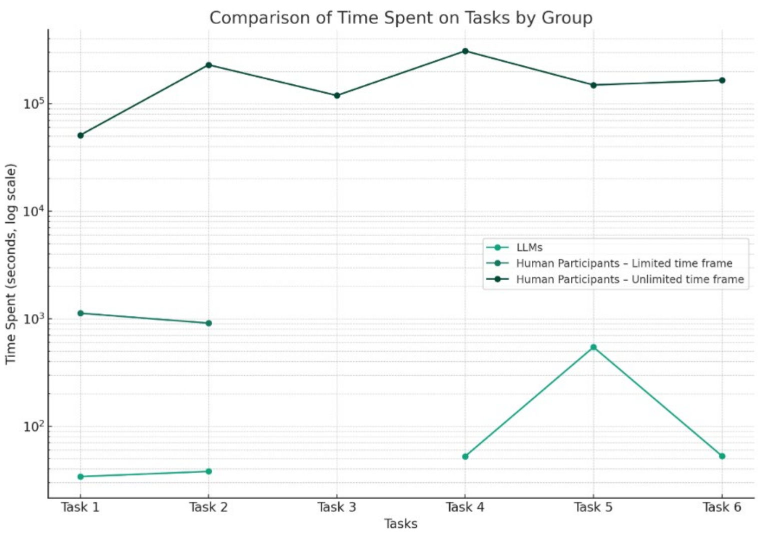

Figure 3: LLMs and Human Participants Time Spent Analysis. Source: [own]

The logarithmic chart above [*Figure 3*] compare the time spent on each task by the separate groups: LLMs (Large Language Models), Human Participants with a Limited time limit, and Human Participants with an Unlimited time limit.

Key Observations:

- The time spent by LLMs is consistently lower across tasks, illustrating their rapid problem-solving capabilities.
- Human Participants with a Limited time limit show a notable increase in time spent on the first two tasks but do not extend to the subsequent tasks.
- Human Participants with an Unlimited time limit demonstrate a significant increase in time spent, especially on the later tasks, indicating the increased complexity and the considerable effort invested in solving these challenges.

3.3 Comprehensive Comparison Across All 15 Tasks

The analysis for all 15 tasks for the groups LLMs and Human Participants with an Unlimited time limit provides a detailed comparison in terms of correctness and time efficiency.

3.3.1 Correctness Analysis

LLMs (Large Language Models)

Exhibited a moderate level of correctness, with rates ranging from 0 % to 33.33 %. The consistency in correctness across various tasks suggests a stable but limited capability in accurately solving the tasks.

Human Participants – Unlimited Time Limit

Demonstrated significantly higher correctness rates, especially in the initial tasks, with rates starting at 88.64% for Task 1 and gradually decreasing in the subsequent tasks. This trend indicates a strong ability to solve the initial tasks accurately, with a decline in success as tasks become more complex.

3.3.2 Time Efficiency Analysis

LLMs

Maintained a consistent and low time spent across tasks, with times ranging from 34 to 544 seconds. The low variance in time spent indicates a uniform approach to problem-solving regardless of task complexity.

Human Participants – Unlimited Time Limit

Showed a significant variance in time spent, with averages ranging from 14.13 to 85.75 hours for different tasks. The wide range in time spent and high standard deviation reflect the diverse strategies and considerable effort invested by participants in solving these challenges.

3.3.3 Comparative Insights

The correctness analysis reveals a clear distinction in problem-solving accuracy between LLMs and human participants. While LLMs maintain a consistent but moderate level of correctness, human participants, given unlimited time, exhibit a higher capability to solve tasks correctly, especially in the earlier challenges. However, as the tasks progress in complexity, their correctness rates decline, yet they remain higher than those of LLMs.

In terms of time efficiency, LLMs demonstrate rapid problem-solving, while human participants exhibit a wide range in time investment, indicating diverse problem-solving approaches and the complexity of the tasks.

3.3.4 Implications

These findings highlight the strengths and limitations of both LLMs and human problem-solving in coding challenges. LLMs offer speed and consistency but lack the higher accuracy rates seen in humans, particularly in complex tasks. The results suggest the potential for a synergistic approach, combining the rapid processing of LLMs with the nuanced and adaptive problem-solving of humans, especially in scenarios requiring high accuracy and understanding of complex problems.

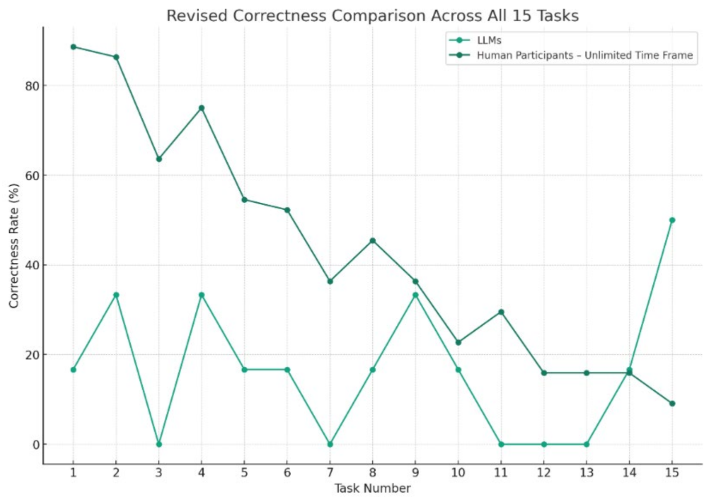

Figure 4: Revised Correctness Comparison Across All 15 Tasks. Source: [own]

The revised plot above [*Figure 4*] present an accurate comparison of the correctness rates for LLMs (Large Language Models) and Human Participants with an Unlimited time limit across all 15 tasks of the 'Advent of Code' challenges.

Key Observations:

- LLMs display a variability in correctness rates across the 15 tasks. The plot shows their performance on the tasks they attempted, with some tasks demonstrating higher correctness rates.
- Human Participants with an Unlimited Time Limit exhibit higher correctness rates in several tasks, with notable performance in the initial tasks as well as some of the later ones. This indicates their ability to solve a range of tasks with varying complexity.

3.4 AI Model Specific Analysis

3.4.1 GPT-4.0 and Custom GPT

- Correctness: Demonstrated a range of correctness from 0 % to 100 % across the 15 tasks, showing a higher level of problem-solving capability with a peak correctness rate of 100 % on several tasks.
- Time Efficiency: The time spent on tasks varied, with an average of 85 seconds for Task 1 and going up to 644.5 seconds for Task 5, indicating an efficient problem-solving time.

3.4.2 GPT-3.5

- Correctness: Displayed a lower correctness rate, with most tasks showing 0 % correctness and a peak of 50 % correctness only on Task 15.
- Time Efficiency: Generally spent less time on tasks compared to GPT-4.0, with times ranging from 7.34 to 16.54 seconds across tasks, suggesting quicker but less accurate responses.

3.4.3 Bard

- Correctness: Like GPT-3.5, Bard also had most tasks at 0 % correctness, apart from 50 % correctness on Task 15.
- Time Efficiency: Time spent was comparable to GPT-3.5, ranging from 4.66 to 21.91 seconds, indicating rapid task completion.

3.4.4 Summary

GPT-4.0 and Custom GPT stand out as the most capable in terms of correctness, handling a wider range of tasks effectively, though taking longer to respond in some cases.

Both GPT-3.5 and Bard showed limited problem-solving abilities with lower correctness rates, although they were quicker in responding.

3.5 Insights on Problem-Solving Strategies

An interesting observation from the study was the diversity in problem-solving strategies. Students employed a range of approaches, from methodical to innovative, adapting their strategies as required. AI models, while consistent in their approach, lacked the adaptability and creative problem-solving exhibited by human participants.

3.6 Implications of Findings

3.6.1 Synergy Between Human Creativity and AI Efficiency

In the realm of coding challenges like the 'Advent of Code', our study reveals critical insights into the interplay between human intelligence and AI models. The findings shed light on the distinct strengths and limitations inherent in both approaches, suggesting a path towards a more integrated and synergistic problem-solving paradigm.

3.6.2 Balancing Strengths and Limitations:

The performance of AI models, particularly the advanced LLMs such as GPT-4.0, is characterized by rapid processing and a consistent approach to problem-solving. However, they often fall short in achieving high correctness rates in complex tasks, where nuanced understanding and contextual interpretation are key. In contrast, human participants, especially with the luxury of unlimited time, demonstrate remarkable adaptability and creativity. They excel in tasks that demand a deeper comprehension of context, highlighting an ability to devise solutions that are not only correct but also innovative.

3.6.3 The Role of Time Constraints:

Our analysis underlines the significant impact of time constraints on problem-solving effectiveness. Participants with limited time limits struggled to maintain high correctness rates, particularly as tasks grew in complexity. This highlights the importance of time as a resource in achieving optimal problem-solving efficacy, especially in environments where accuracy is as crucial as speed.

3.6.4 Potential for Hybrid Problem-Solving Models:

The contrasting capabilities of humans and AI models pave the way for a collaborative approach in coding challenges. By amalgamating the computational efficiency of AI with the nuanced, adaptive problem-solving of humans, it becomes possible to achieve solutions that are both efficient and highly accurate. Such a hybrid model would be particularly valuable in scenarios demanding a blend of speed, accuracy, and creative insight.

3.6.5 Advancements in AI Models:

The progression from GPT-3.5 to GPT-4.0 illustrates ongoing enhancements in AI's problem-solving abilities. Nevertheless, the need for human-like creativity and contextual understanding remains a vital aspect that AI has yet to fully replicate.1

3.6.6 Future Research Directions:

This study not only elucidates the current capabilities of AI and human intelligence in algorithmic challenges but also sets the stage for future explorations. Investigating the collaborative potential of human and AI intelligence can lead to groundbreaking solutions that capitalize on the strengths of both. It opens new avenues in programming and algorithm development, where the creative problem-solving capabilities of humans are synergized with the relentless efficiency of AI.

4. Conclusion

The research embarked on a journey to explore the dynamic interplay between human intelligence and artificial intelligence (AI) in the sphere of algorithm development. Through a meticulous examination of the 'Advent of Code' challenges, insightful contrasts in the problem-solving approaches and efficiencies of human participants and neural network models have been uncovered, including the latest iterations of Large Language Models such as GPT-4.0, GPT-3.5, and Bard.

The analysis revealed that while AI models, particularly GPT-4.0, exhibit remarkable computational speed and consistency, they often fall short in achieving high correctness rates, especially in tasks that demand a deeper contextual understanding and creative reasoning. On the other hand, human participants, particularly those not constrained by time, demonstrated a superior ability to adapt and apply more effective problem-solving strategies, achieving higher accuracy in solving complex challenges.

The study illuminates the unique strengths and limitations of both human and AI-driven algorithm development. It underscores the potential of a synergistic approach that harnesses the computational prowess of AI with the creative and adaptive problem-solving skills of humans. Such a collaborative model could significantly enhance efficiency and accuracy in algorithmic challenges, paving the way for innovative solutions in various technological and scientific fields.

The research not only contributes to the understanding of the current capabilities of AI-assisted programming but also opens avenues for future exploration into the untapped potential of human-AI collaboration. It sets a foundation for further investigations into hybrid problem-solving models where the ingenuity of human intelligence is augmented by the relentless efficiency of AI.

In conclusion, the "Efficiency Divide" study highlights the evolving narrative of programming and algorithm development in the modern era, highlighting the complementary roles of human creativity and AI's computational efficiency. As we progress further into the age of AI, the fusion of human and machine intelligence stands as a promising frontier for future innovations.

References

Advent of Code. (2023). Stats - Advent of Code 2023. Accessed on: January 24, 2024 [Online]. Available at: https://adventofcode.com/2023/stats

Lee M., Srivastava M., Hardy A., Thickstun J., Durmus E., Paranjape A., ... & Liang P. (2022). Evaluating human-language model interaction. Accessed on: January 17, 2024 [Online]. Available at: https://arxiv.org/abs/2212.09746

Advent of Code. (2023). About - Advent of Code 2023. Accessed on: January 05, 2024 [Online]. Available at: https://adventofcode.com/2023/about

OpenAI. (2024). Introducing GPTs. OpenAI Blog. Accessed on: January 22, 2024 [Online]. Available at: https://openai.com/blog/introducing-gpts

Arora, S., Narayan A., Chen M.F. (2022). Ask me Anything: A Simple Strategy for Prompting Language Models. Accessed on: January 12, 2024 [Online]. Available at: https://arxiv.org/abs/2210.02441

Si Ch., Gan Z., Yang Z. (2022). Prompting GPT-3 To Be Reliable. Accessed on: January 14, 2024 [Online]. Available at: https://www.researchgate.net/publication/364438038_Prompting_GPT-3_To_Be_Reliable

Olsen, A.L. (2005). Using Pseudocode to Teach Problem Solving. Accessed on: January 14, 2024 [Online]. Available at: https://www.researchgate.net/publication/234778626_Using_pseudocode_to_teach_problem_solving