PART A: Reviewing research papers about game-playing AI systems

Introduction to the field of game-playing AI

Game-playing AI, when simply put, means Artificial Intelligence systems that play games where they can challenge human beings. Game-playing AI aims to solve real-life complex problems by learning from games containing engaging and challenging situations. The AIs can learn decision-making skills that allow them to succeed in games, which can also be applied in real-life situations.

The techniques commonly used are Reinforcement learning, Large-language models, and Machine-learning based programs.

Descriptions of three papers about game-playing AI What are the researchers trying to achieve?

Article 1[From Motor Control to Team Play in Simulated Humanoid Football]

The problem addressed is how low-level motor instructions support coordinated movements and how the humanoids can work as a team to play the game. Imitation, reinforcement, and population-based learning are the techniques used to address the problem. Imitation learning is used for low-level motor skills where available knowledge on the domain is used.

Reinforcement learning trains individual humanoids to acquire basic skills while allowing all humanoids to work together to showcase a complete game. Population-based training allows the agents to compete against each other and learn to improve themselves, and discover new tactics to win the game as a team.

The research is evaluated based on the behavioral statistics of the agents when the game is played without any human intervention. Counterfactual policy analysis is also used where environmental conditions are controlled to view how this affects the agents' responsiveness. Probe tasks evaluate how the agents react to particular game scenarios under strict conditions*(Liu et al., 2022)*.

The researchers managed to observe that within a day, the agents managed to learn about the game, and they were able to improve on ball possession and movement. After another six hours of training, the players could get back up eight percent of the time they fall. After training even longer, the agents can display teamwork and collaborative play. They also improved on passing and the distance they could pass the ball. The positioning of the players also improves constantly as the players compete against each other, developing a competitive behavior naturally. This causes an improvement in synchronized team play. With further training and selecting the best agents, the researchers built an AI to master the football game.

No open-source code or data is provided. However, the agents' behavior is recorded in a video that can be accessed to understand better the agents' performance and how they improved over further training and refinement.

Article 2[VOYAGER: An Open-Ended Embodied Agent with Large Language Models]

The problem addressed in this article is how we can use a Large Language Model-based and incorporated long-term learning agent in Minecraft to discover the game world constantly, gain a wide variety of skills and make new findings without human involvement. The agent consists of three main components. Firstly, a fully automated curriculum that improves discovery. Secondly, an ongoing developing skill library composed of executable code for obtaining complicated behaviors and storage. Lastly, an iterative prompting system was created to put together execution errors, responses from the environment, and self-verification for developing the system further*(Wang et al., 2023)*.

The techniques allow the agent to communicate with GPT-4 through situational learning and commands. This enables the production of the automated curriculum, which considers Voyager's current position and its development in exploration. The skill library is a foundation for gaining new knowledge and evolution. It is adapted from the universality of programs, generability, and interpretability(*Wang et al., 2023)*. Every skill is defined by a running code consisting of actions for successfully finishing a particular mission, which the automatic curriculum recommends. A genuine iterative mechanism is also used to improve the system, where it is implemented as a new skill and installed by indexing its description. If a skill is to be obtained, the library is queried with spontaneous task plans and commands from the environment. The agent can gain new knowledge and continue improving when the skill library is regularly expanded and improved. Lastly, the iterative prompting mechanism uses a code interpreter to collect commands from the environment and implementation errors. These are then integrated into the GPT-4's prompt for code improvement in the following round of code generation. This process is executed continuously till the mission's completion. The new skill is then added to the skill library and request a new goal from the automatic curriculum.

The research is evaluated by comparing the agent with the baseline models available such as ReAct, Reflexion, and AutoGPT. They were analyzed in terms of how effective they were in discovering new areas, the amount of distance they were able to cover, how fast where they were able to build new items from the skills they gained, and zero-shot generalization capacity to simple tasks in the game environment*(Wang et al., 2023)*.

The researchers achieved their goals largely as the agent can unlock the wooden and stone levels faster than the baselines and even reach the diamond level easily. The agent can also travel even longer distances than the other baseline models. To analyze zero-shot generalization, the backlogs of the agents are cleared and placed in unknown environments. GPT-4 is used for both the Voyager and AutoGPT, where the missions are broken into small tasks. The Voyager can still constantly solve all the missions, while AutoGPT cannot solve any missions under fifty commands. The skill library is also very effective in improving Voyager's work and can also improve the performance of AutoGPT. This shows that the skill library can be used as an effective tool that can be implemented as a plug-in in other models to improve agents' performance.

The researchers provided the algorithms used for designing the agent in the article to allow us to improve further on the existing framework and develop the agents even further.

Article 3[Google DeepMind's game-playing AI just found another way to make code faster]

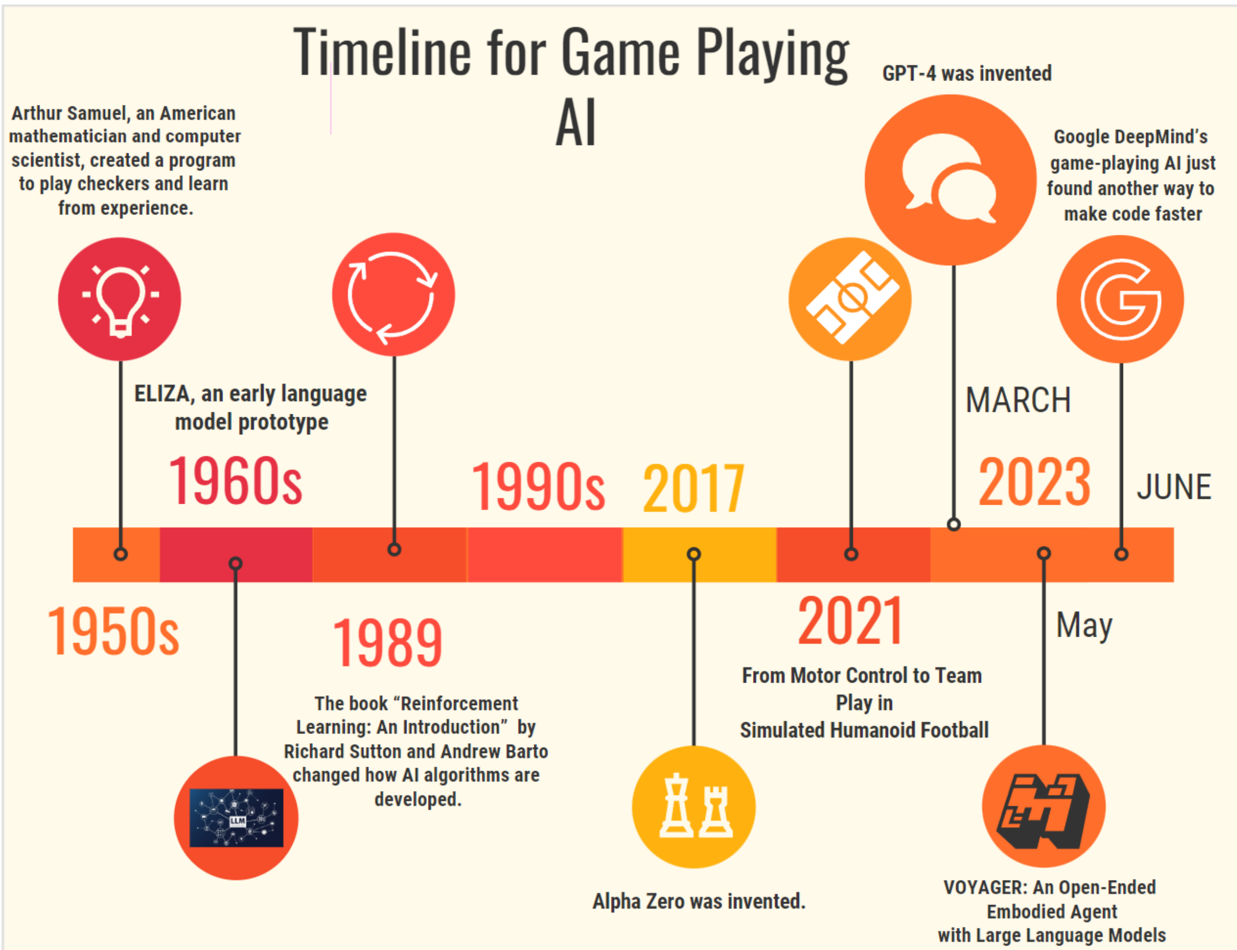
AlphaDev can sort a list seventy percent quicker than current models. It is also able to aid in improving the performance of an important algorithm present in cryptography by thirty percent.
AlphaDev is an upgraded version of AlphaZero, a reinforcement learning model by Deepmind that was created to ace games such as chess and Go. The researchers designed a game that aims to solve the problem of discovering a quicker algorithm and then train AlphaDev to win the game. Simply put, the game can be won by AI if it can discover a new algorithm that is faster and more accurate than available algorithms. AlphaDev must

run through large amounts of possible moves, and it also uses assembly, a programming language that provides important commands on the movement of numbers in the system's chip. The language also simplifies algorithms by breaking them down into simpler steps. AlphaDev uses assembly to add new commands to its algorithm but was unsuccessful in creating working algorithms. However, it was able to come up with winning moves after a short period through continuous training and refinement.

The researchers evaluated their results by considering how AlphaDev developed a successful algorithm using just seventeen instructions to sort a list containing three items*(citation)*. It could also complete the task around seventy percent faster than regular models. However, it is difficult to compare AlphaDev with existing algorithms as it only uses a limited number of instructions provided in assembly. AlphaDev also has a disadvantage where the learning rate decreases with longer commands.

The team did manage to achieve the goal of creating a faster algorithm to sort a list. However, they could not make the AI work effectively when the algorithms get longer. There is still room for improvement and more work to refine AlphaDev. The team did propose to use C++ language to train AlphaDev in tackling longer algorithms, but it is still a work in progress. No open-source code is provided for us to view or refer to.

Creating a timeline of developments in game-playing AI



Discussion of ethics of game-playing AI

Ethics were not discussed in any of the three papers I have chosen. Ethical issues are involved in games such as Voyager and AI playing football. One of the main issues is Privacy. When AIs are implemented into games, it gathers users' behavioral data, such as their choices and actions. It is not explicitly stated how this information will be stored, implemented, and distributed despite the AI collecting them only for research purposes.

Another concern is addiction. Users must spend extended periods playing these games where they might develop harmful gaming habits, significantly affecting their mental and physical health.

In the case of AlphaDev, the AI may create biases in its algorithm as it finds shortcuts to solve the problem without considering all the instructions provided by the programming language. When implemented in real-life situations, this might affect the way the AI analyzes the problem and finds an appropriate solution that is not immoral.

Statement on the reliability of the references chosen

The papers are reliable as researchers from top corporations such as DeepMind, NVIDIA, Caltech, and Stanford write them. DeepMind, NVIDIA, and Caltech are top technological companies specializing in researching Artificial intelligence and experimenting more. Researchers from Stanford are also mentioned as advisors who have vetted and read through the papers before publication.

Word Count: 1,462 words

Reference list

[1]Liu, S. et al. (2022) From motor control to team play in simulated humanoid football - science, From motor control to team play in simulated humanoid football. Available at: https://www.science.org/doi/10.1126/scirobotics.abo0235 (Accessed: 14 July 2023).

[2]Wang, G. et al. (2023) An open-ended embodied agent with large language models, Voyager. Available at: https://voyager.minedojo.org/ (Accessed: 14 July 2023).

[3]Heaven, W.D. (2023) Google Deepmind's game-playing AI just found another way to make code faster, MIT Technology Review. Available at: https://www.technologyreview.com/2023/06/07/1074184/google-deepmind-game-ai-alphadev-algorithm-code-faster/ (Accessed: 14 July 2023).

[4]Professional Infographic Maker: 10,000+ templates (no date) Venngage. Available at: https://venngage.com/ (Accessed: 14 July 2023).

[5]Armin Norouzi, Ph.D. (2023) The brief history of large language models: A journey from eliza to GPT-4 and google bard, Medium. Available at: https://levelup.gitconnected.com/the-brief-history-of-large-language-models-a-journey-from-eliza-to-gpt-4-and-google-bard-167c614af5af (Accessed: 14 July 2023).

[6]Sajid, H. (2023) A comprehensive overview of large language models, Wisecube AI. Available

at:https://www.wisecube.ai/blog/a-comprehensive-overview-of-large-language-models/ (Accessed: 14 July 2023).

[7]Frąckiewicz, M. (2023) Ethical implications of AI-enhanced video games: Reality, representation, and responsibility, TS2 SPACE. Available at: https://ts2.space/en/ethical-implications-of-ai-enhanced-video-games-reality-representation-and-responsibility/#:~:text=Many%20AI%2Denhanced%20games%20require,more%20invasive%20forms%20of%20surveillance. (Accessed: 14 July 2023).

[8]Frąckiewicz, M. (2023a) Ai Ethics in gaming: Balancing immersion and responsibility, TS2 SPACE. Available at: https://ts2.space/en/ai-ethics-in-gaming-balancing-immersion-and-responsibility/ (Accessed: 14 July 2023).

[9]Admin (2023) Who invented reinforcement learning? The AI Blog. Available at: https://aiblog.co.za/ai-faq/who-invented-reinforcement-learning (Accessed: 14 July 2023).