The Desire for Intelligent Machinery

In his seminal paper, *Intelligent Machinery*, Alan Turing poses a question regarding the possibility for machines to show intelligence. He introduces the idea of man as a machine, drawing from the psychologies of human intelligence and applying it to the training of machines through various methods, including the pleasure-pain system. Finally, however, he concludes that while discipline can be achieved through the universal Turing machine, we have not yet uncovered the secret of initiative, that which encompasses the residue of human intelligence. I argue that at this point in time, we have begun to uncover the secrets of intelligent machinery through the revolution of proof search and other machine-assisted search methods, as noted in the lecture *Machine Assisted Proofs* by mathematician Terence Tao. However, while we are leaning into machine intelligence, we must be aware that "the isolated man does not develop any intellectual power," (23) and search for new techniques must be regarded as carried out by the human community as a whole, rather than purely by individuals—or individual machinery. As IBM stated in 1979, a computer cannot be held accountable, therefore a computer should never make decisions.

The belief that machinery could attain intelligence stems from the capacity of machines to mimic certain aspects of human functions. Consider the microphone, which captures sound akin to the human ear, or the camera, which mirrors the function of human vision. These inventions demonstrate how machines can replicate sensory experiences, effectively serving as extensions of human perception. It is even possible to produce robots which can gesture in much the same way humans can move their limbs. These gestures not only mimic physical actions but also convey meaning and intention, blurring the line between human and machine behaviors. This prompts us to view humans as complex biological machines, suggesting that the fundamental principles governing human cognition and behavior could potentially be replicated through machinery. Logically, then, we might think of emulating the human brain itself. Just as machines can simulate sensory inputs and physical movements, they could replicate neural networks and other processes of human intelligence. This concept forms the foundation of artificial intelligence, which drives efforts of creating machines capable of learning, reasoning, and adapting to their environment in ways analogous to human thought processes.

When we regard humans as machines, we must acknowledge that, like machines, we are susceptible to various forms of interference that shape our behaviors and perceptions. Turing outlines two kinds of machine interference—screwdriver interference, where parts of a machine are removed and replaced, and paper interference, where information is communicated to a machine, causing it to alter its behavior. Regarding the latter, we consider a machine modifiable when it is possible to alter the behavior of the machine very radically through this process of interference. As interference occurs, the machine is undoubtedly changed; "it is in this sense that interference modifies a machine," (12).

Turing is interested in machines with little interference. To create intelligent machinery, one must begin with a machine that has a very basic capacity for operations and is subsequently subjected to controlled interference to simulate the process of education. In this framework, the initial machine possesses a limited capacity for complex operations, akin to a blank slate awaiting input. It lacks the sophisticated capabilities associated with intelligence but serves as a foundation upon which learning can occur. The process of interference mimics education in the human context and involves exposing the machine to external stimuli, information, and tasks designed to shape its behavior. Through these interventions, the machine undergoes modifications and gradually acquires new skills, knowledge, and cognitive abilities. This mirrors the process of learning observed in humans, where exposure to educational materials, instruction, and practice leads to the acquisition of skills and understanding. The ultimate goal of this approach is to guide the machine's development until it reaches a level of proficiency where it can reliably follow commands and perform tasks autonomously. This approach emphasizes the importance of systematic and structured learning processes. Instead of attempting to instill machines with preconceived knowledge or abilities, we acknowledge the necessity of allowing machines to learn and evolve iteratively, much like a human learner. Therefore, we can state the goal of modifying unorganized machines into universal machines in the same way that a child's brain is organized by education and interaction with society.

The pleasure-pain system represents a significant departure from traditional approaches to machine training. As Turing observes, the training of a human child relies heavily on a system of rewards and punishments, wherein desirable behaviors are reinforced through pleasurable experiences while undesirable behaviors are deterred through experiences of pain or discomfort. Building upon this insight, Turing proposes a method of organizing machines using these two fundamental interfering inputs. The pleasure-pain system represents a framework for organizing and training machines, rooted in principles of reinforcement learning and behavior modification. Central to the pleasure-pain system is the recognition that machines, like humans, can be incentivized to learn and adapt their behavior based on the consequences of their actions. By associating certain behaviors with pleasurable outcomes and others with painful consequences, machines can be guided towards desired outcomes. At its core, the system operates on the premise of the unorganized machine, whose configuration is characterized by two fundamental expressions—character-expression and situation expression. These expressions encapsulate the current state of the machine's character and its situational context, serving as the foundation upon which subsequent actions and behaviors are determined. At any given moment, the character and situation of a machine, together with input signals, determine its character and situation in the next moment. This process of input-output mapping forms the basis of the machine's learning and adaptation capabilities. Pleasure interference within this system serves to stabilize the machine's character, effectively reinforcing behaviors associated with pleasurable outcomes. When the machine receives pleasurable stimuli in response to its actions, its character becomes fixed, making it more likely to repeat those actions in similar contexts. In contrast, pain interference disrupts the machine's character, inducing changes in its features and behaviors. When the machine experiences painful stimuli due to incorrect actions or deviations from desired outcomes, its character undergoes modification, prompting it to adjust its behavior in future iterations. Pain and pleasure stimuli thus serve as powerful feedback signals, guiding the machine towards desired behaviors and away from undesirable ones. Pain stimuli occur when the machine's behavior deviates from expected norms or leads to unfavorable outcomes, signaling the need for corrective action. Conversely, pleasure stimuli are elicited when the machine performs correctly or achieves desired outcomes, reinforcing the behaviors associated with success. Over time, the repeated exposure to pleasure and pain stimuli gradually molds the character of the machine, steering it in the desired direction. Through a process of trial and error, the machine learns to associate specific actions and behaviors with corresponding outcomes, refining its responses to optimize performance in diverse contexts.

Turing's pleasure-pain system, in drawing from human psychology, provides us with an innovative way to instill discipline in machinery. This approach to machine learning enables the systematic organization of behavior while empowering machines to adapt and respond to dynamic environments. However, unlike humans, machines trained within this framework lack the intrinsic drive to explore new possibilities or seek novel solutions autonomously. Instead, their actions are predominantly reactive, driven by external stimuli and conditioned responses. As we continue to advance artificial intelligence, addressing this limitation will be crucial for unlocking the full potential of intelligent machines in addressing complex and unforeseen challenges.

We have outlined the ways in which Turing recognizes that machines can be disciplined to follow instructions and perform tasks. True intelligence, however, requires something more elusive—initiative. In Turing's view, initiative represents the ability to go beyond mere programmed responses and engage in creative problem-solving, adaptability, and independent decision-making. One avenue through which machines might demonstrate initiative, according to Turing, was through the ability to generate novel proofs. In his paper "Computing Machinery and Intelligence", Turing suggests that the ability to discover new proofs, rather than merely executing predetermined algorithms, could be indicative of genuine intelligence. In his lecture, "Machine Assisted Proofs", mathematician Terence Tao emerges as a key figure in the exploration of intelligent machinery. Through his examination of proof search and other machine-assisted search methods, Tao illuminates the potential of employing computational techniques in mathematical inquiry.

In his exploration of proof search, Terence Tao introduces us to Satisfiability (SAT) solvers—algorithms capable of performing logical deductions within specific sets of hypotheses and generating proof certificates. One notable example Tao brings to light is the Boolean Pythagorean Triples Theorem, a classic problem in number theory. This theorem asserts the existence of a particular arrangement of binary valuables that satisfies certain logical constraints, analogous to the Pythagorean theorem in traditional geometry. Proving this theorem eluded mathematicians for decades due to the sheer computational complexity involved in exploring all possible configurations. However, SAT solvers, through sophisticated search strategies and optimization techniques, have been able to systematically explore the vast solution space of the Boolean Pythagorean Triples problem. This proof required 4 CPU-years of computation and generated a 200 terabyte propositional proof. In principle, this is a finite computation and *could* be approached by human mathematicians, it is unfortunately computationally unreasonable. While we have no human-derived proof, then, we know that this theorem is true purely because of the intelligent machinery—the SAT solver—that was able to reasonably compute all possible answers to the triples problem.

There are new, emerging ways in which we can develop machine-assisted proofs. Three novel approaches which Tao explores in his lecture are formal proof assistants, machine learning algorithms, and large language models. Despite their individual strengths and weaknesses, there is immense potential for these tools to complement each other in the pursuit of machine-assisted proofs. By integrating all three, mathematicians can leverage the strengths of each approach while mitigating their respective weaknesses.

Formal proof assistants represent a rigorous and systematic approach to proof construction, wherein mathematical arguments are encoded in a formal language and subjected to rigorous verification by computational systems. The formalizing process begins with the creation of a human-readable "blueprint" of the proof in question. This blueprint synthesizes the formal and informal proof in an interactive fashion. Through this synthesis, the program allows for large-scale collaborations that could not be done with traditional proof search methods for various reasons. First, it does not require high-level trust because the machine will confirm the correctness of the work as it is uploaded in the system. Second, this format allows for division of labor. Someone who is an expert in formal proof building can focus on formalizing the blueprint in software, while someone else, such as a mathematician, can work on the human-readable portion of the proof. Therefore, one does not need to possess all the skills necessary to complete the proof formalization process, but can employ people with different skills in a large-scale collaborative format in order to construct a proof. These tools excel at ensuring the correctness and validity of mathematical proofs, providing mathematicians with a reliable framework for establishing theorems with precision. However, the process of manually constructing formal proofs can be labor-intensive and time-consuming. It takes longer to formalize proofs than to create human-readable proofs, but there is the benefit of correctness.

On the other hand, machine learning algorithms offer a more data-driven approach to mathematical reasoning, employing vast datasets and computational power to identify patterns and potential proofs. These algorithms are able to detect underlying structures and relationships in mathematical data, enabling mathematicians to uncover new insights that may have eluded human intuition. A notable example of ML application is in the study of knots and hyperbolic invariants. Researchers trained a neural network using a database of over two million knots, which enabled the network to predict the signature from the hyperbolic invariants—descriptors used to characterize the properties of knots in three-dimensional space—with remarkable accuracy. What makes this discovery particularly intriguing is that the connection between the signature and hyperbolic invariants was not anticipated by existing mathematical theory. However, despite the successful prediction by the neural network, it was not clear exactly what the connection was, only that there existed one. To gain insight into this relationship, researchers conducted saliency analysis to determine exactly how sensitive the neural net relation was, and revealed that only three out of the hyperbolic invariants tested significantly influenced this connection. By visually examining scatterplots of the signature against these three invariants, researchers were able to formulate a conjecture about their relationship. While subsequent numerical experiments disproved their initial conjecture, a modified version was proven through rigorous mathematical analysis. This example demonstrates the potential of artificial intelligence in uncovering hidden patterns and relationships in complex mathematical systems. Through the computational power of neural networks in conjunction with mathematical analysis, researchers were able to make significant strides in understanding the connections between knots and hyperbolic invariants, ultimately advancing our knowledge in this area of mathematics.

Finally, the most high profile development in the field of proof search has been large language models, such as the incredibly popular Chat GPT. These models are trained on massive corpora of text data and aim to generate coherent and contextually relevant mathematical proofs and explanations. While large language models are more able to capture the nuances of human language and reasoning, they tend to lack the formal rigor and verifiability associated with formal proof assistants. Additionally, despite their remarkable achievements in tasks like language translation, text generation, and even creative writing, LLMs often struggle with even the simplest of arithmetic operations. Nevertheless, explorations are underway for innovative ways to harness the potential of LLMs in mathematical problem-solving. One such approach involves integrating LLMs with other computational tools, such as Wolfram Alpha, which excel at performing complex mathematical computations and providing detailed explanations. By leveraging the strengths of both LLMs and computational tools, we can overcome the arithmetic limitations of LLMs and extend their capabilities to more sophisticated mathematical tasks. Another strategy for enhancing the mathematical shortcoming of LLMs involves testing their output through formal proof assistants. Through this task, we can ensure the accuracy and validity of mathematical statements produced by these models. While we are not yet able to use

LLMs for formal proof search, they can still serve as a way to stimulate mathematical thought. Interactions with language models like GPT could provide us with creative and unconventional methods for tackling long standing mathematical problems through prompting discussions and posing open-ended questions. Additionally, conversing with language models could aid us in locating relevant concepts and allow researchers to identify connections between disparate mathematical concepts, facilitating interdisciplinary collaboration. The interactive nature of language models empowers researchers to iteratively explore and refine their ideas.

In conclusion, computational tools are increasingly being leveraged to assist human mathematicians in a number of ways, extending far beyond brute-force case checking. They serve as valuable assistants, offering insights, suggestions, and computational power to augment the problem-solving abilities of mathematics. However, it is essential to recognize the distinction between the role of computers as tools and the role of human mathematicians as creators and interpreters of mathematical knowledge. While computers excel at performing certain tasks with speed and efficiency, they still lack the creative intuition and deep understanding that characterize human mathematical inquiry. As such, while computers can contribute significantly to mathematical research, they should not be regarded as replacements for human mathematicians. Returning to the concept of cultural search, it is clear that computers play a vital role as collaborators in the endeavor of mathematical exploration. They contribute to the diversity of approaches and perspectives on challenging problems and thus enrich the discourse on mathematics. However, it is the unique blend of human insight, creativity, and ingenuity that ultimately drives progress in mathematics. In essence, while computers are invaluable tools in the mathematician's toolkit, I believe they should be viewed as assistants rather than autonomous problem solvers. Embracing this perspective ensures that we harness the full potential of computational technology while preserving the essential role of human intellect in the pursuit of mathematical knowledge.