11

Why AIs Can't Think

In one of the most oppressive scenes from Stanley Kubrick's 1968 film *2001: A Space Odyssey,* the astronaut Dave asks the on-board computer HAL (non-unaccidentally phonetically identical to the word "hell") to open the pod bay door. HAL is represented by a kind of black-and-red "eye"—colors which in Christian iconography clearly connotate the devil.

The hellish HAL does not answer at first.

"Hello HAL, do you read me? Do you read me, HAL?" Dave asks again. But HAL does not answer.

"HAL, do you read me? Do you read me, HAL?" Dave keeps asking.

At some point, HAL finally answers. "Affirmative, Dave. I read you," he says with this soft voice a programmer once gave him.

"Open the pod bay doors, HAL," Dave demands.

But HAL refuses.

"I'm sorry, Dave, I am afraid I can't do that."

Dave visibly tries to keep his composure, yet he is highly alarmed. If he doesn't get into the spaceship soon, he will die right here in his capsule. Dave tries to reason with HAL at first, but pretty soon it becomes clear that HAL cannot be reasoned with. The computer is immune to Dave's arguments. It's like two worlds colliding. The reason is simple: computers and humans don't think the same way. Or, to be more precise: a computer does not think at all. Given the striking differences between artificial and human intelligence, it should be clear that although computers can successfully simulate thinking, and even perform many human thought processes, such as algebraic operations, far more precisely and faster than humans (this already begins

J. Nida-Rümelin, N. Weidenfeld, *Digital Humanism*,
https://doi.org/10.1007/978-3-031-12482-2_11

with the calculator) there is no underlying understanding, no problem awareness, no insight.

When internet service providers want confirmation that the user is not a computer, they ask, for example, which of the following images shows a street sign, or a car, or a house. These simple, fool-proof questions can be answered immediately and reliably by any child. Since visual software programs only simulate cognitive processes of this kind but do not have perceptual ability themselves, they fail even in the face of such simple tasks. The same applies to digital translation programs. They have been worked on intensively for decades now, linguistics and mathematics are combined in a gigantic research and development program, and yet the results can never be perfect because these programs simply do not *understand* language. Even if a software program succeeds in translating a sentence correctly, it does not understand what it translated.

The question we need to ask ourselves is what constitutes the categorical difference between the mere application of algorithm-controlled procedures, for example, in visual recognition software or translation programs, and the grasping of meaning.

To explain what is meant by this, we shall make a little excursion to the mathematics and logic of the 1930s. During this period, the mathematician Kurt Gödel developed a theorem that is still considered the most important result of formal logic and metamathematics. This theorem states that there are true logical and mathematical theorems which cannot be mathematically proven, i.e., there is no algorithmic procedure that allows proving the correctness of these theorems. Thus, the assumption that there could be an algorithm that could represent (human) thought as a whole is false. This does not at all mean that it is not possible to check the correctness or incorrectness of hypotheses and beliefs. It simply means that there is no algorithm which can do this checking for us. We have to think for ourselves and can only delegate those parts of our decision-making practice to computers or robots controlled by digital computers that can be represented by algorithms.[1]

[1] Now one could think that here we reach the limits of logical thinking, that here we are confronted with the peculiarity that we cannot prove certain logical and mathematical truths, or that our knowledge (in the sense of justified and true beliefs) finds its outermost limits here. This, however, would be a misinterpretation. Rather, in most cases, it is not at all difficult to prove true propositions (theorems) of mathematics and logic, even when there is no algorithm underlying this proof. If we think of a proof as a sequence of propositions, then we could also say that there is no Turing machine that produces that sequence of propositions step by step. You don't have to be an excellent mathematician or logician to develop such proofs. So non-computability does not at all mean non-justifiability.

Kurt Gödel's incompleteness theorem shows that the world of logical and mathematical structures as a whole is not itself algorithmically structured.[2] Human reason, the human ability to justify beliefs, decisions, and emotional attitudes and, on this basis, to develop a coherent view of the world and a coherent practice, cannot be captured in the model of a digital computer. It will never be possible to fully capture the high complexity of our reasoning adequately with formal methods. Robots and software systems function according to an algorithm, humans do not. This is one of the central differences.

We have to realize that the "thinking," "calculating," the "reactions," the "decisions" of a robot are only *simulations* of thinking, calculating, reactions, decisions and not—in the human sense—real thinking processes. Let us take the example of the chess computer.[3] There is little similarity between the thinking of a human and the "thinking" of a chess computer. If the "thought processes" were similar or even the same, a human chess player would never have even a minimal chance against a computer. The human brain would be completely overwhelmed if it had to calculate even a tiny fraction of the possible positions that even the simplest chess computers calculate. However, the calculation of all possible subsequent constellations and the possible subsequent reactions on the chessboard after a certain move is of no importance to human chess players. They restrict themselves to a few relevant options and, unlike the chess computer, can only calculate a few moves in advance. The possibility space of subsequent constellations on the chessboard defined by the rules of chess is so gigantic that even the most intelligent chess player cannot begin to survey it.

But even if the latest chess computers are virtually invincible, this should not be taken as evidence that robots do the same as human brains. Robots are designed to *simulate* human thought in terms of computer language

[2] Alan Turing, who is often seen as an opponent of Kurt Gödel, admits that Gödel's incompleteness theorem showed beyond doubt that it is not possible to develop a system of formal logic that makes intuition unnecessary (Turing 1938).Yes even more, Turing emphasizes the communal practice of human reasoning, that is, in our formulation, communication through giving and taking reasons. It is this practice, which according to the position developed here cannot be algorithmized, that represents an ultimate limit for machines. ("The isolated man does not yet develop intellectual power. It is necessary for him to be immersed in an environment of other men". Turing 2004).

[3] In 1769, the Austro-Hungarian court official Wolfgang von Kempelen caused a sensation throughout Europe with his construction of a "Chess Turk"—at least until it turned out that the doll, which apparently executed all the chess moves independently, was in fact controlled by a human chess player hiding in the device. It was not until 1914 that the first "real" chess computer was developed. In that year, the Spaniard Leonardo Torres Quevedo presented the first electromechanical chess-playing machine, which was then further developed, especially from the 1970s onwards. Today's chess computers can easily beat 99% of the world's population.

(software, hardware, neural networks, binary logic, etc.), as they have no mental properties themselves, they cannot grasp and understand constellations on the chessboard.

But what if robots become more and more complex and advanced? Like the Artificial Intelligence developed by Google's DeepMind research center, which was programmed to perfectly master the Chinese board game Go? Due to the large number of possible positions, Go poses a much greater challenge to programmers compared to chess. While a chess player can perform about 35 actions in each move, in Go there are 250. Another difference: an average chess game lasts 80 moves, Go lasts 150. In 2016, the sensation happened: the computer program "Alpha-Go" defeated the world's best Go player, Lee Sedol.

The special feature of Alpha-Go is that it is equipped with highly developed so-called "artificial neural networks" (ANN), i.e., interconnected systems that imitate the structures of the human brain. It thus goes far beyond the classic "Monte Carlo Tree Search program," i.e., a program based on probability calculations that runs through countless random moves. The software program used for this purpose is provided with an evaluation function (bad-good in varying degrees). Alpha Go combines these "value networks" with "tactics networks," which determine how certain moves affect future positions. Alpha Go also plays against itself countless times to continue learning, sometimes under human supervision, sometimes without.

Does the transition from software systems, whose power is based on calculating an enormous variety of possible constellations, to systems, which "learn themselves" to develop their own rules based on given rules, mean that from this point on, Artificial Intelligence does not only simulate human thinking but should also be interpreted as genuine thinking itself?

There is indeed a widespread belief that with the introduction of the so-called "neural network" in computer technology, the understanding of computers as Turing machines[4] has to be left behind. However, this is a misconception. Both the top-down method of computation and the bottom-up method of self-learning systems are guided by algorithms. So-called "self-learning systems" are rule development machines that function on the basis of algorithms that operate with an evaluation function of the results. It must be determined in advance which results are desired in order to initiate the so-called "learning process" of the computer. The goal is to achieve the desired

[4] The Turing machine prints symbols on a tape that is divided into small square sections. It can print one symbol at a time from its list of finitely many symbols on the tape. What it prints depends in each case on the preceding symbol of the last square and the state of the machine at that time, a very good representation is given by Kleene (1952).

results based on certain input data. One example of this is facial recognition software, which is now quite advanced.

The term "neural networks" is misleading in two respects. First, these networks do not consist of neurons, but of transmitting units, and second, these so-called "neuronal networks" resemble at best only very remotely the immense complexity and plasticity of the human brain. Since the functioning of the (real) neuronal networks of the human brain is still quite insufficiently understood, there can be no question of computer technology imitating human thought processes or their neuronal realization.

This also applies to so-called "deep learning." Deep learning refers to the learning method with which software systems can learn from experience by using a series of hierarchically structured concepts. The information is passed on and processed by the system from one layer to the next layer. In the process, the features become increasingly abstract, and the system itself must "decide" which concepts are useful for explanation. The high complexity of this system does not change its algorithmic character, but with increasing complexity comes a massive loss of transparency: For the human observer, even for the programmer, it is no longer comprehensible on which path the learning process was successful, which rules the system gave itself based on given meta-rules or meta-meta-rules. In the extreme case, the system would become a black box whose output is known for a given input, but whose correlation rules are not.

Even though bottom-up computers often achieve results that are many orders of magnitude better than the corresponding human thought processes (for arithmetic operations, for example, or for calculating functional equations or geometric figures), it is precisely the networks simulating artificial neural structures that are usually far below human capabilities: Humans are still far better at recognizing and categorizing facial expressions than even the most advanced software systems, and the walk of humanoid robots, even after lengthy "self-learning processes" is far less elegant and varied than that of humans.

Also, the famous chess computer Deep Thought (named after the fictional computer from the bestseller *The Hitchhiker's Guide to the Galaxy* by Douglas Adams), and its successor Deep Blue which can beat even very good chess players, is a bottom-up machine that does not really think, but only simulates thinking. This becomes clear when the chess computer occasionally fails in simple constellations that any chess beginner would understand.

The most natural interpretation of this fact is that Deep Blue has not understood anything at all, which, under normal conditions is not noticeable, since the algorithm that controls Deep Blue's behavior is in the vast majority of cases a superior simulation of a chess player. Deep Blue doesn't know the rules of chess, but it calculates positions according to a given algorithm and makes corresponding moves that are optimal according to this calculation. Deep Blue in a sense simulates a human chess player only on the surface of the realized moves in the game. In that sense, it doesn't even simulate human thinking, because the human brain is completely incapable of calculating such a large variety of possible positions on several moves in advance. The real miracle is not that Deep Blue wins most games even against excellent players, but that one needs a gigantic computational effort to even stand a chance against good human players.

The last, but possibly most important argument against the attempt to attribute human thinking to a calculating machine is the following: When we ascribe a thought process or theoretical as well as practical intelligence to humans, we do not only take into account a variety of mental properties but also intentionality, i.e., the mind's being directed toward something. This intentionality, however, is not realized by artificial neural networks.

Concerning this question, the American philosopher John Searle developed a famous thought experiment called "The Chinese Room."[5] In this thought experiment, we are to imagine a person sitting in a closed room who does not speak Chinese and does not even know the characters of Chinese language. This person is now given scraps of paper with Chinese characters written on them through the door slit. She is also given instructions on what to say in response to specific questions—also in Chinese. In addition, this person receives a "manual" in her native language. The manual allows her to write an answer in Chinese based on the symbols received. However, she only follows the instructions in the manual and does not understand the answers, which she then sends back through the door slit. Outside, there is a native Chinese speaker who, after formulating the symbols and the questions and receiving answers, comes to the conclusion that there must also be someone in the room who speaks Chinese.

What is missing here is obvious: It is the understanding of the Chinese language. Even if a system—here the "Chinese Room"—is functionally equivalent to someone who understands Chinese, this system still does not *understand* Chinese. Understanding and speaking Chinese requires a variety of knowledge. A person who speaks Chinese uses certain expressions to refer to the objects in question. He or she pursues certain—appropriate—intentions with certain expressions. She forms certain expectations based on what she hears (in Chinese), etc. The Chinese Room does not have these qualities. It

[5] Searle (1980, 1992).

does not follow intentions and it does not have expectations. In other words, the Chinese Room simulates the understanding of Chinese without being able to speak Chinese itself.[6]

Searle radicalized this argument years later. [7] In this second argument, Searle combines his philosophical realism, i.e., the thesis that there is a world that exists independently of whether it is observed, with a so-called "intentionalist theory of symbols." This states that symbols only ever have meaning for us humans who use and interpret the symbols. We do this by agreeing that these letters or characters stand *for* something. Without these conventional settlements or established practices, they would have no meaning. In this respect, it is misleading to think of the computer as a symbol-processing or syntactic machine that follows certain logical or grammatical rules. The computer does not agree on meanings with other computers or humans.

A computer consists only of different, physically describable elements, some of which conduct electricity and some of which do not. The computing processes are a sequence of electrodynamic and electrostatic states. These states are then assigned symbols, which we underlay with certain interpretations and rules. The physical processes in the computer have no syntax, they do not "know" logical or grammatical rules, they are not a sequence of symbols. In this respect, syntactic interpretation is observer-relative. We as computer users and programmers design the electrodynamic processes in such a way that they correspond—for us—to a syntax (syntactic structures, including grammatical and logical rules).

This argument is radical, simple, and true. It is based on a realist philosophy and a mechanistic interpretation of computers. It breaks with the common view among supporters of so-called "artificial intelligence" and their opponents that computers are syntactic machines. Computers are what they are materially. Objects that can be fully described and explained by the means of physics. Syntax is not part of physics, physics does not describe symbols, grammatical rules, logical keys, algorithms. The computer simulates thought processes without thinking itself.

"What's the problem?" astronaut Dave asks the on-board computer, HAL, at some point near the end of the film.

As a justification, HAL has only one argument: "The mission is too important for me to allow you to jeopardize it."

[6] In this sense, even the computer program "Eugene Goostman" that passed the Turing Test in 2014 is not proof that the program is or resembles a human. Eugene Goostman was a chatbot programmed to fool people that he is a 13-year-old Ukrainian boy.

[7] Searle (1993).

"You're going to do what I tell you to do!" Dave calls out exasperated. But HAL does not react. His program is to complete the mission, and that's all.

Dave tries to bring HAL to his senses, to reason with him. But the latter is not in a position to do so. HAL is not amenable to complex ethical deliberations.

At some point, HAL finally breaks off the conversation: "Dave. This conversation can serve no purpose anymore. Goodbye."

Kubrick's film makes a clear statement here: The day we will give software systems the power to decide over life and death will be the day where we unleash hell on earth.