Croatian Journal of Philosophy
Vol. XX, No. 58, 2020

Are people smarter than machines?

PHIL MAGUIRE, PHILIPPE MOSER and REBECCA MAGUIRE
National University of Ireland, Maynooth, Ireland

Recent progress in artificial intelligence has led some to speculate that machine intelligence may soon match or surpass human intelligence. We argue that this understanding of intelligence is flawed. While physical machines are designed by humans to simulate human rule-following behaviour, the issue of whether human abilities can be emulated is not well-defined. We outline a series of obstacles that stand in the way of formalizing emulation, and show that even a simple, well-defined function cannot be decided in practice. In light of this, we suggest that the debate on intelligence should be shifted from emulation to simulation, addressing, for example, how useful machines can be at particular tasks, rather than deliberating over the nebulous concept of general intelligence.

Keywords: Artificial intelligence, Turing test, Church-Turing thesis, technological singularity, simulation, Turing machines.

1. *Introduction*

Could a human-made machine ever surprise its creator, by taking initiatives of its own? According to Cristianini (2016), this is a question that has been asked for centuries, resulting in a variety of answers. Arguably the first computer programmer, Ada Lovelace knew where she stood on this issue: "The Analytical Engine has no pretensions whatever to originate anything", she stated in 1843. "It can follow analysis; but it has no power of anticipating any analytical relations or truths".

And yet, 173 years later, a computer program developed just over a mile from her house in London beat Lee Seedol, a 9-dan master of the game Go. None of AlphaGo's programmers could ever hope to defeat Lee Seedol, let alone defeating their own program. According to Cristianini (2016), the software has learned to do things its programmers can't do and don't understand. The machine learning techniques used by AlphaGo are becoming widespread in the field of AI. Whereas in the past the idea of a "learning machine" might have sounded like a con-

tradiction, it now seems reasonable to speak of physical machines that are flexible, adaptive, or even curious (Cristianini 2016).

Given these recent breakthroughs, some commentators have suggested that machine learning will continue to improve to the point where it surpasses human ability in many domains. At the Future of Humanity Institute's conference on machine intelligence in 2011, Sandberg and Bostrom (2011) conducted an informal poll eliciting the views of participants. The median estimate for the emergence of human-level machine intelligence was the year 2050 (see also Müller and Bostrom 2016).

We argue in this paper that the idea of AI somehow surpassing humanity constitutes a misrepresentation of the nature of intelligence. The goal of machine learning is not to recreate intelligence or even to define it, but instead to show that many tasks that were previously assumed to require human intervention can be successfully automated. The pertinent question is not whether machines are going to overthrow humanity in a technological singularity (e.g. Bostrom 2014), but how resistant different aspects of human behaviour are to simulation.

In order to make this case we highlight a series of obstacles that lie in the way of formally defining the concept of emulating human intelligence. We then show that even simple, well-defined functions cannot be decided in practice. Returning to the question of whether people are smarter than machines we conclude that, rather than grappling with the concept of general intelligence, philosophers should instead focus on anticipating the utility that machines might provide on particular constrained tasks.

2. *Problems with the question of emulation*

Building on earlier work by Kurt Gödel, the theory of computation was independently discovered from different perspectives in 1936 by Alonzo Church and Alan Turing. Church's version was based on the λ-calculus, while Turing's was based on what is now known as the Turing Machine. In his 1936 article, Turing presents the idea of a Universal Turing Machine (UTM), which is capable of simulating any other Turing Machine. The key ingredients for this breakthrough are:

1) the idea of capturing general recursive functions (a.k.a. computable functions) in the form of a simple model for symbol manipulation (a.k.a. Turing Machines)
2) the philosophical position that general recursion captures all effective methods, a position now known as the Church-Turing thesis

With these two ingredients, all effective processes can be enumerated. This enumerability supports the concept of universal computation, as it allows a single "Universal Turing Machine" (UTM) to read in a description of any other effective method to be simulated, as represented

by the machine's index in the ordering of Turing Machines. A UTM is thus capable of simulating any process that is effectively calculable.

This result seems to open the door to human-level AI: a single physical machine can be developed which can, when given the appropriate program, simulate every effective process conceivable. Could a UTM running a specially-designed program therefore emulate the 'program' running in the human brain? We identify several obstacles that lie in the way of formalizing and then deciding such a question.

2.1 *We don't know what it means to build a Turing machine in practice*

The UTM is an abstract mathematical idea. Physical machines are engineered to offer only finite precision, rather than the potentially unbounded precision and memory space required by Turing's definition. Thus, physical machines merely *simulate* the behaviour of a genuine Turing machine. This issue goes beyond the lack of an infinite tape, it concerns the very mechanics of the device: we simply can't be sure that a physical machine will continue to compute properly without making mistakes at some point in the future due to unforeseen engineering limitations. In the words of Wittgenstein (RPP I 1096), Turing's 'machines' are, actually, "humans who calculate". Turing (1948/9) himself clarifies that human behaviour sets the standard for his concept: "A man provided with paper, pencil and rubber, and subject to strict discipline, is in effect a universal machine."

Physical machines are engineered by humans to *simulate* human computing abilities with a certain level of fidelity. They are not intended to emulate the standard of computation benchmarked by humans. By 'emulate' we mean to match or exceed human capacity, as opposed to 'simulate', which involves a finite level of success at imitation:

> The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer. (Turing 1950)

Also:

> Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner. (Turing 1950b)

The dual interpretation of the word 'machine', as in Turing machine (human abstraction) versus computing machine (physical device), can lead to confusion. The 'machine' in the term 'Turing machine' refers to the idea of strict rule following without imagination. It does not refer to a physical device. Humans compute in a way which is captured by the abstract idea of a Turing machine, whereas electronic devices are engineered to simulate that behaviour. Human-built machines represent only a finite amount of engineering calibration carried out by a relatively small set of humans. Whereas human consensus sets the standard for computation, electronic devices merely simulate that abil-

ity to a finite degree of precision, one which is continually improved by developments in technology (see Maguire and Maguire 2018).

Because Turing machines are a mathematical ideal, it remains unclear how such a concept could be represented by a physical object in practice.

2.2 *We can't formalize what it means for a UTM to be universal*

The cornerstone of computation is the concept of the stored program, the idea that no function is special, that every function can be represented as data inputted to a single UTM. However, this intuitive foundation depends crucially on the Church-Turing thesis (see Copeland 2002). The Church-Turing thesis asserts that a function on the natural numbers is computable by a human following an algorithm (ignoring resource limitations) if and only if it is computable by a Turing machine. In other words, it states that the idea of a Turing machine captures everything there is to the notion of a human following the step-by-step instructions of an algorithm.

The physical form of the Church-Turing thesis in even more restrictive. It states that a Turing machine captures every act of algorithmic rule-following that a human can achieve even when exploiting the use of exotic physical processes, such as black holes or quantum systems (see Cuffaro and Fletcher 2018; Earman 1993; Kieu 2004). Thus, the physical Church-Turing Thesis could be false without humans necessarily being able to demonstrate a superior ability using only pen and paper.

The concept of universal computation (i.e. the idea that a UTM exhausts the set of effectively computable functions) depends on the Church-Turing thesis. If the Church-Turing thesis turns out to be false, then Turing machines would lose their privileged position: there would exist logical processes that could not be simulated by any given Turing machine, but which could be computed by a human using some other rule-following process. There might not be any automaton capable of computing all the new functions, in which case the concept of universality would be lost.

As soon as we accept that an automaton is capable of universal computation, or that it supports a Turing-complete language, we are relying on the Church-Turing thesis. Nevertheless, we have no *proof* that the Church-Turing thesis is true, and, according to Turing (1954), have no aspirations of ever discovering such a proof (cf. Black 2000; Dershowtiz and Gurevich 2008). By definition, the thesis seems to be outside the scope of proof, because it speaks about the set of effective methods, and the process of proof-checking is in that set. Although there are no known counter-examples, and different formalisms converge towards the same result, the Church-Turing thesis is not the type of statement we aim to prove. It exists as an informal statement in natural language, not in a form that could be processed by a Turing machine. So, is the thesis 'true'?

Certainly, there has been no refinement in the notion of effective method since 1938, when Kleene refined Turing and Church's (1936) method by applying it to partial functions. In that sense, there is convincing evidence that it is *hard* to identify a stronger notion of effective method than that provided by Church and Turing. And yet, it is not possible to guarantee that, at some point in the future, a new development in the understanding of human rule-following abilities will reveal a limitation in the Turing machine's functionality that vitiates its putative property of exhausting the set of effective methods. We simply don't know. Although the concept of universal computation is intuitively convincing, its existence is not something that has been formally proved.

Although Church and Gödel were happy to accept the Church-Turing account of effective method as a *definition* (according to Gödel "... the correct definition of mechanical computability was established beyond any doubt by Turing"), Emil Post, who in 1936 delivered an alternative model of computation, was vociferous in his opposition. According to him, "to mask this identification under a definition hides the fact that a fundamental discovery in the limitations of mathematicizing power of Homo Sapiens has been made and blinds us to the need of its continual verification". Post hoped to publish a series of "wider and wider formulations ... The success of the program would for us, change this hypothesis not so much to a definition or to an axiom but to a natural law" (Post 1936).

Turing adopted the middle ground, accepting computation's strong intuitive foundation, while at the same time acknowledging that the thesis would always remain unproved. In 1954 he remarked: "The statement is...one which one does not attempt to prove. Propaganda is more appropriate to it than proof, for its status is something between a theorem and a definition."

In sum, the Church-Turing thesis remains an informal statement, not a mathematical one. It cannot be fully formalized, and consequently it cannot be processed by a computing machine. It is a sophisticated thesis expressed in natural language, on which the universality of a UTM depends, yet it lies beyond the scope of a UTM.

This presents another obstacle to the emulation of human intelligence. The recognition of emulation depends on the recognition of universality, which itself hinges on the truth of a sophisticated thesis in natural language, which cannot be formally addressed by a Turing machine.

2.3 *We can't formally address the potential existence of human abilities beyond computation*

Even if we could somehow formally confirm the Church-Turing thesis, there might still be some human abilities which remain beyond rule-based computation. In other words, a UTM might have some limita-

tions that humans are able to 'appreciate' in some nebulous sense that is itself beyond automation. Any claim that machines can emulate human abilities has to deal with this possibility.

For example, in his 1936 paper, Turing identified the existence of a well-defined problem which is beyond computation (i.e. an uncomputable problem). This allowed him to resolve in the negative the Entscheidungsproblem, the question of whether there exists an algorithm to decide whether a given statement in first order-logic is provable or not. Turing showed by contradiction that no such algorithm is possible. He imagined taking an automaton which decides if an algorithm will halt or not, and then feeding it a description of itself (now made possible by the enumerability of effective methods). This creative leap allowed him to precisely define an object, known as the halting language, which cannot be produced by any effective method whatsoever.

The issue here is that no Turing machine can ever 'know' that the halting language exists. To do that it would have to be able to appreciate that there is something it cannot do, without trying to do it. Thus, it *seems* that humans hold a privileged perspective over Turing machines, insofar as we 'know' that the halting language exists. For example, Lucas (1961) argued that humans can look and see that a given machine's Gödel sentence is true, meaning they can always do something that the machine cannot. This argument has been further developed by Penrose (1994) to show by contradiction that human abilities could never be formalized to the point at which a Gödel sentence becomes discernible.

Similarly, even though the halting language cannot be constructed or represented in nature, it seems to be defined for the human reader in a finite number of symbols by Turing's 1936 article. If humans can indeed appreciate such a definition, then they are capable of recognizing the idea of an object that no computer can ever represent.

In his 1938 PhD thesis, carried out under the supervision of Church, Turing makes clear his view that the human mind has an intuitive power for performing uncomputable steps beyond the scope of a Turing machine:

> Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two faculties, which we may call intuition and ingenuity. The activity of the intuition consists in making spontaneous judgments which are not the result of conscious trains of reasoning...In consequence of the impossibility of finding a formal logic which wholly eliminates the necessity of using intuition, we naturally turn to non-constructive systems of logic with which not all the steps in a proof are mechanical, some being intuitive.

After the Second World War, Turing's view on the role of intuition in reasoning appears unchanged. In a 1948 report to the National Physical Laboratory, Turing again clarifies that mathematicians' ability to decide the truth of certain theorems appears to transcend the methods available to any Turing machine:

> Recently the theorem of Gödel and related results...have shown that if one tries to use machines for such purposes as determining the truth or falsity of mathematical theorems and one is not willing to tolerate an occasional wrong result, then any given machine will in some cases be unable to give an answer at all. On the other hand the human intelligence seems to be able to find methods of ever-increasing power for dealing with such problems, 'transcending' the methods available to machines.

In his last article published before his death in 1954, Turing again emphasises the role of intuition beyond effective method. He argues that Gödel's theorem shows that 'common sense' is needed in interpreting axioms, something a Turing machine can never demonstrate:

> The results which have been described in this article are mainly of a negative character, setting certain bounds to what we can hope to achieve purely by reasoning. These and some other results of mathematical logic may be regarded as going some way towards a demonstration, within mathematics itself, of the inadequacy of 'reason' unsupported by common sense.

Human intuition is an ability that resists formalization or description. It is not possible to verify the emulation of human abilities if we cannot even represent what those abilities are.

2.4 *Even simple, well-defined functions are undecidable in practice*

In sum, merely formalizing the question of whether physical machines can emulate human intelligence is fraught with great difficulty. And yet, even if it could somehow be formalized, the question would still not be a useful one, because it couldn't be answered in practice.

Let's assume that it's somehow possible to build a true physical Turing machine. Let's assume that we somehow 'know' for sure that Turing machines are capable of universal computation, and that all aspects of human behaviour can be described in terms of that computation. Even with all of these assumptions, it can be shown, using an argument from theoretical computer science, that the question of emulating a given program remains *undecidable*, since, in practice, even the simplest functions are undecidable from their output.

Below, we provide a modification of the use theorem (see Odifreddi 1992) to show that no finite set of interactions is sufficient for deciding what process, whether computable or uncomputable, is behind the behaviour of a black-box system: properties of functions cannot be decided in practice based on their output. No matter how many questions are asked, it is not possible to know for sure what is behind the output of a black-box system. The argument is related to that of Gold (1967), who showed that any formal language that has hierarchical structure capable of infinite recursion is unlearnable from positive evidence alone. It is also related to Rice's theorem (1953), which shows that all non-trivial semantic properties of programs are undecidable (in the case of Rice's theorem access is given to the program code itself, rather than its output).

More formally, let O be an observer, A a set of strings over some finite alphabet Σ, and f: $\Sigma^* \rightarrow \{0,1\}$, our black-box, be a Boolean function.

O can adaptively ask finitely many queries $f(x)$=? (O has access to A), after which O decides whether f computes the set A, i.e. $f(x)=A(x)$ for every x.

The following standard argument shows every observer is wrong on some function (i.e. the past behaviour of the black-box cannot be used to decide its future behaviour).

For any observer O, and any set A, there is a function f: $\Sigma^* \rightarrow \{0,1\}$ *such that O is wrong on f.*

Proof.
Let O be as above, A be a set. If O rejects all functions (i.e. thinks all functions do not compute A) then O is wrong on f, where $f(x)=A(x)$ for every x. So let g be accepted by O. O queries g on finitely many strings $x_1,x_2,\ldots, x_n$. On all the strings $x_1,x_2,\ldots, x_n$, g is equal to A, otherwise O is wrong about g. Choose y different from $x_1,x_2,\ldots, x_n$, and construct f: $\Sigma^* \rightarrow \{0,1\}$, by letting $g(x)=f(x)$ for all $x \neq y$, and $f(y)=1\text{-}A(y)$. f does not compute A, because f is different from A on input y. Because f equals g on inputs $x_1,x_2,\ldots, x_n$ (the ones queried by O), O will make the same decision about f as about g, i.e. O decides that f can compute A. By construction of f, O is wrong.

In sum, this result shows that not only is a finite interaction incapable of deciding intelligence, it is not even capable of deciding any function whatsoever. Even an oracle with access to the halting language could not produce behaviour which reliably separates it from a simple Turing machine. Past behaviour is never sufficient for deciding whether a system is doing something smart. A computable process can mimic an uncomputable one up to any finite duration of interaction.

The question of emulating human intelligence thus holds no utility in practice. No finite set of interactions can always be relied on to expose the lack of intelligence of any given machine. If a black-box system has not yet made a mistake, there is no way to tell whether or not it will make any mistakes in the future. Thus, the propensity to make mistakes at some stage *does not matter*.

According to Turing (1948): "the condition that the machine must not make mistakes ... is not a requirement for intelligence". At the infinite limit, mistakes are inevitable, but in practice those mistakes can be pushed back as far as one wants. Turing (1947), in his earliest surviving remarks concerning AI, points out that this would allow machines to play very good chess:

> This...raises the question 'Can a machine play chess?' It could fairly easily be made to play a rather bad game. It would be bad because chess requires intelligence. We stated... that the machine should be treated as entirely without intelligence. There are indications however that it is possible to make the machine display intelligence at the risk of its making occasional serious mistakes. By following up this aspect the machine could probably be made to play very good chess.

Rather than dismissing the idea that humans are ultimately smarter than machines, Turing (1950) is instead highlighting the lack of practical significance of such an idea: in the physical world there will always be some machine which is up to the job of simulating intelligence to a required finite length before making any mistakes:

> There would be no question of triumphing simultaneously over all machines. In short, then, there might be men cleverer than any given machine, but then again there might be other machines cleverer again, and so on.

It should be noted that reducing human behaviour to the computation of a function is already a very strong modelling assumption, even before the issue of emulation is tackled. The observable behaviour of some physical systems, such as chaotic deterministic systems, cannot be described by the computation of any function (see Longo and Paul 2011). Applying a functional description to biological individuals would no doubt be close to impossible.

In practice then, emulation is a thoroughly useless idea, or in Turing's (1950) words, an idea "too meaningless to deserve discussion". Attributing intelligence to any being or object with certainty is an undecidable issue at best. Whatever intelligence is, it's not something that depends on confirmation. So what is it? In the remainder of the paper we examine alternatives to emulation.

3. *Emulation is not a useful concept, but simulation is*

Thus far we have highlighted why the question of a machine emulating human abilities has no utility. But if the emulation of intelligence holds no utility, does this imply that intelligence is itself a useless idea? What, if anything, does the concept of intelligence imply in practice? One possible conclusion is that intelligence has no discernible real-world effects whatsoever, having nothing to do with behaviour.

This is the attitude adopted by Professor Jefferson in his 1949 Lister Oration (which Turing was responding to in his 1950 article): "Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain—that is, not only write it but know that it had written it."

Here, Jefferson is arguing that behaviour alone is never sufficient for providing evidence of intelligence. Instead, we must 'know' what words mean and 'feel' emotions. Because such properties can never be represented symbolically, there is no possibility of any system, human or otherwise, evidencing its intelligence in practice. Intelligence and behaviour have no relationship at all.

But this doesn't seem right. Intuitively, what we mean by 'intelligence' is something useful. Our human abilities let us achieve things in the real world that simple rule-following systems could not. It seems as though we can quickly and reliably identify intelligence through the communication of symbols. For example, this article is only a few pages

long, yet (we hope) it strongly suggests an intelligent origin. It seems feasible that a finite signal beamed from a distant solar system could convince us that it harbours intelligent life. We could never be absolutely 100% sure, but it seems plausible that there exist signals that could lead us to be very, very confident.

Indeed, all the communication that has ever taken place between human beings can be summarized as a finite string of symbols. Human communication, of course, relies not just words, but also gesture, voice tone, facial expression, body language and context. Assuming a continuous high fidelity recording of what an individual sees and hears, all of this information could be translated into a finite set of 1s and 0s. If intelligence could not be evidenced in practice through finite interactions, it would preclude humans from identifying each other as intelligent, reducing us to solipsists.

It seems that in order for the concept of intelligence to be a meaningful and useful one, there must be some practical means of identifying and engaging with intelligent systems in the real world. Having realised this, Turing (1950) remarks "I am sure that Professor Jefferson does not wish to adopt the extreme and solipsist point of view. Probably he would be quite willing to accept the imitation game as a test."

Intelligence does something in practice. Although it cannot be used to *decide* the intelligent origin of a signal (i.e. choose yes or no with absolute certainty), it seems as though it can be used to detect the footprint of intelligence with high confidence. According to Aaronson (2006), "people regularly *do* decide that other people have minds after interacting with them for just a few minutes...there *must* be a relatively small integer n such that by exchanging at most n bits, you can be reasonably sure that someone has a mind" (see also Harnad 1992).

With this in mind, Turing (1950) makes clear that the interesting question is not whether machines can emulate humans (an undecidable proposition at very best), but *how difficult* it will be to build useful machines that simulate human behaviour closely, for extended periods of time. Specifically, the questions about intelligence that can be meaningfully asked and answered are those concerning how resistant different human abilities are to simulation.

Turing switches the focus from emulation to the simulation of intelligent behaviour, describing the idea of an imitation game, a 'test' which sets human behaviour as the standard to be simulated for a finite duration. The goal is for machines to confound the heuristics that people typically rely on for detecting signals of intelligent origin. Hodges (2009) succinctly expresses this idea: "operations which are in fact the workings of predictable Turing machines could nevertheless appear to the human observer as having the characteristics of genuine intelligence and creativity".

To be clear, Turing's test is not a test for *deciding* intelligence. Turing (1950) never once refers to machines 'passing' his test; the test is not intended to provide evidence of anything beyond the ability of a machine to do well at that test for a finite period of time, hence the notion of 'passing' doesn't hold any particular significance. The imitation game merely provides a vehicle for quantifying how resistant human behaviour is to simulation (albeit, an unreliable one). If a machine passes one test, we do not deduce anything further about the abilities of that machine, because there is no guarantee whatsoever that it will pass another test. Instead, we conclude that the test is not as hard as we thought it was, that it is perhaps no longer a strong test for intelligence. The test is not intended to address the question of whether machines can emulate intelligence (i.e. that they could simulate human behaviour perfectly for any length of time). Instead, Turing's (1950) contribution is to take a question "too meaningless to deserve discussion" (i.e. "Can machines think?" / can machines emulate human abilities?) and to transform it into a meaningful question that can be addressed in practice (how resistant are human abilities to simulation?).

Turing seeks merely to establish the possibility of "satisfactory performance" at the imitation game over a finite period (i.e. finite simulation); not perfect performance, nor the idea that satisfactory performance is a perfect predictor of subsequent perfect performance. He never goes beyond claims for the finite simulation of intelligence: "My contention is that machines can be constructed which will simulate the behaviour of the human mind very closely" (Turing 1951).

4. *Designing a good test*

Some have interpreted Turing (1950) as suggesting that infinite testing is required to establish intelligence, spread over an infinite length of time (e.g. Harnad 1992). Again, Turing's focus is not on establishing that machines can emulate human thinking, a concept which he describes as "meaningless". Instead, he is speculating on the difficulty of identifying reliable tests for discriminating human intelligence. Even if humans have intuitive abilities beyond machines, it may be difficult to demonstrate such abilities *in practice*. How hard is it to identify a reliable test for intelligence?

Let's consider the question of "what is a good test"? A test is of finite length. Applying it to an object yields results that enable inferences to be drawn about that object. Somehow, the results hold significance for other aspects of the object, beyond those which have been directly tested. One could say that the test succeeds in succinctly 'characterising' the object through a finite set of investigative results.

For example, students are asked to sit tests to reveal how much they know about a particular subject. Because of the short duration, it is not possible to ask them every question that could possibly be asked. Instead, questions are chosen cleverly so that responses can be relied

on to draw inferences about students' ability to answer all the other potential questions which haven't been asked. A good test allows the tester to make inferences about future behaviour based on past behaviour.

Of course, a particular student might get lucky on a test. They might fortuitously (or by cheating) have learned off the answers to the exact questions which came up, but no others. Thus, as previously argued, a test can never *decide* whether a student fully understands a subject. What a cleverly crafted test can do is offer a very high level of confidence that the student would have answered other questions correctly. Past behaviour can allow us to predict future behaviour with high confidence.

What are the properties of a good test that would lead us to have such confidence? In short, a good test is one for which there is no easy strategy for passing it, other than full mastery of the subject. For a start, there should be no way for students to get a copy of the test in advance, or predict what will be on it so that they can learn off the relevant responses without understanding the subject deeply. In addition, the test should be well diversified, bringing together material from many different areas of the subject. For instance, the answers should draw on different aspects of understanding, and not betray a simple pattern which would allow them to be all derived using the same technique. Finally, successive answers should be integrated with each other, rather than addressing separate chunks of knowledge which could be learned off independently. When one answer builds on the next, the only way to do well is to understand everything.

These criteria for test reliability can be summarized as follows: the content of the test should be *random* relative to the set of questions that could potentially be asked, and also internally integrated, so that questions cannot be answered independently of each other. If we follow these criteria, it seems the difficulty of passing the test can increase exponentially relative to its length.

If test questions were leaked in advance, then machines would only need to hardcode the appropriate responses to ensure success. How can we ensure that test questions are as unpredictable as possible? Turing's (1950) idea is to hand this responsibility over to human judges. Given that they can rely directly on their own intelligence (whatever that is), questions derived by humans on the fly have the potential to be hard to answer. In addition, human judges have the greatest ability to integrate subsequent questions into the preceding conversation, so as to ensure there is no trivial algorithm for computing the relationship from input to output.

Of course, this only applies in the best case scenario, when human judges choose the most challenging questions conceivable. But how do we know which questions are reliable indicators of intelligence?

Although intelligence might give us the ability to pass convincing tests easily, it does not necessarily give us the ability to easily find, generate or recognize good tests. Given a particular test, how can we *know*

that there is no simple program that quickly computes the answer? For example, the Winograd schema challenge (e.g. Levesque 2014) currently poses great difficulty for machines, yet we have no guarantees that developments in AI over the next few years will render such problems obsolete.

Whenever researchers put forward what intuitively appears to be a challenging test for AI, such as playing chess, nobody knows for sure how hard it really is. Problems that are assumed to require deep insight can end up having relatively simple mechanical solutions. For example, Hofstadter (1980) erroneously predicted that no dedicated program would ever defeat a chess champion, because playing the game well constituted a test for general intelligence:

> "Do you want to play chess?" "No, I'm bored with chess. Let's talk about poetry". That may be the kind of dialogue you could have with a program that could beat everyone. (Hofstadter 1980)

Turing's concept of a test is not intended as a once-off decider of emulation. Instead, it represents the idea of a never-ending battle to establish the superiority of human intelligence over rule-following. Turing believed that it would be a battle in perpetual retreat, with supposedly reliable tests continuing to fail:

> It is customary, in a talk or article on this subject, to offer a grain of comfort, in the form of a statement that some particularly human characteristic could never be imitated by a machine...I cannot offer any such comfort, for I believe that no such bounds can be set. (Turing 1951)

Inevitably, if a Turing-style test is run using laypeople, the programs that get furthest will be those that exploit the weaknesses of human psychology. People who aren't trained in AI can be more easily fooled. Thus, rather than being inferred from the length of questioning, resistance to simulation could be quantified by unrestricted open competition, with significant prize money awarded to expert machine-exposing teams. Turing seemed to assume that the tester would be at the level of an informed graduate of Oxford or Cambridge (McDermott 2015). Either way, Turing's opinion was that coming up with new tests that reliably separate people from machines was going to get harder quite quickly.

Once a test is found to be passed by a machine, then the test is busted. It can no longer be relied on to provide evidence of intelligence. As soon as a machine succeeds in defeating a test, researchers go back to the drawing board to develop a harder test. The process never ends. In the same way that it is not possible to decide intelligence, it is not possible to decide the reliability of a test for intelligence. Tests must themselves be tested, in an unending cycle of doubt.

Even though the question of emulating human abilities is useless, the practical issue of evidencing an ability gap between machines and humans is becoming more and more challenging. This explains why Turing (1950) was upbeat on the imminent prospect of artificial intel-

ligence. The behaviours that were intuitively assumed to be reliable tests in 1950, such as playing good chess, or engaging in convincing conversation, had never been exposed to machine-scrutiny before, making Turing quite confident that they would not hold up for long. This confidence is evident in his prediction that by the end of the 20th century people would "be able to speak of machines thinking without expecting to be contradicted".

5. *Are people smarter than machines?*

Who is better at the game of Go, humans or physical machines? Although there will probably never again be an individual human that can defeat or improve on the top Go-playing program, we expect that humanity as a whole will continue to overthrow every reigning program by constantly designing better and better ones.

For instance, humans have already created an improved computer program that is capable of beating AlphaGo, called AlphaGo Zero. Developed within 2 years of its predecessor, it beat the original version of the program 100 games to 0 (see Silver et al. 2017). If humanity as a whole retains the ability to improve on the world's top Go-playing software, then humanity must know something about the game of Go that the software does not. While the individual AI developers who together created AlphaGo Zero are individually beaten by their collective creation, they can, when working together, find ways to improve on the state of the art. Technology and AI offer a way for humans to combine and leverage their collective engineering prowess into a single system whose efforts can be focused on a specific problem which is beyond the understanding of any single person.

The main advantage that a program has over an individual human is being able to concentrate the historical wisdom provided by many different people over a lengthy period of time, and to apply it quickly. Machines can be faster, cheaper, more reliable, more durable, and can hold greater memory. However, this 'brute force' does not equate to emulation. Brute force can surpass human ability over short runs, but in the longer run humans have the ability to innovate superior algorithms for performing the same task even more quickly.

For example, just because current chess programs can beat any grandmaster in the world at chess does not mean that computers are better at chess than humanity as a whole. We can say that they simulate the intelligent playing of chess well, under certain conditions, for a certain period of time. But we still cannot show that a program *emulates* human ability at chess.

The output of any human-built machine simply reflects the stored historic work of humans, which always involves a *finite* amount of effort drawn from a potentially unbounded set. The possibility always remains for humans to carry out even more work, and build an even better machine, which more closely simulates human intelligence.

While highly engineered machines can 'simulate' human ability to avoid mistakes for a long time, such performance is never sufficient to rule out the possibility of some bug that was too rare to be anticipated. For this reason, at no point in the future will humans recognize the behaviour of a human-built machine as the ultimate authority for what counts as logically correct. Human-built machines will always have human errors embedded in their makeup that their original builders missed, but that the hardware and software engineers of tomorrow can fix. Consequently, humanity does not learn about logic by observing the activity of human-built machines.

Granted, an individual human can make a mistake relative to the standard held by a larger group of humans, or temporarily to a machine, but the whole of humanity cannot make a mistake relative to a machine. While physical machines may provide useful information to one person in a particular context, they never provide information to humanity as a whole: human-built machines merely represent a store of humanity's logical and engineering effort, which can then be reused and applied to novel problems.

From this perspective, we can see that the idea of physical machines surpassing humanity is nothing more than a clever trick. Physical machines can certainly impress individual humans, but only by recycling and cleverly blending the stored historical wisdom of larger groups of humans.

And yet ... any claims that humanity might make to ultimate superiority over machines reflect nothing more than useless intuitions. As noted by Turing (1950), assertions of the mind's superiority are "without any sort of proof". Intuitively we seem to 'know' that people are smarter than machines, but in practice that means nothing.

6. *Identifying useful questions for AI*

Interpretations of Turing's (1950) work have focused strongly on the idea of a specific challenge that, once passed, has significant implications. For example, Warwick and Shah (2015) claim that the Eugene Goostman chatbot machine "became the first to pass the Turing Test, as set out by Alan Turing, on unrestricted conversation". Turing's article has often been interpreted either as being supportive of functionalism (e.g. Searle 1980), or of advocating a trite, deeply flawed test for evaluating the intelligence of artificial systems through the process of imitation (e.g. French 2012). Shieber (1994), for instance, interprets Turing (1950) as making the claim that "any agent that can be mistaken by virtue of its conservational behaviour [for] a human must be intelligent" (see Copeland 2003, for further examples). Hayes and Ford (1995) go so far as to interpret Turing as proposing "a test of making a mechanical transvestite" and state that "Turing's vision from 1950 is now actively harmful to our field".

Although the specific test described by Turing is no doubt dated, we have argued that his article is not focused on emulation. Because humans cannot even express what it would mean for a physical machine to emulate human abilities, the question is useless. Instead, Turing was speculating about what physical machines of the future would be able to accomplish in practice. He hinted at, not a specific challenge, but a general thought experiment, involving the hypothetical *simulation* of human intelligence by imaginable computers. Although it currently seems as though we can quickly and reliably identify human intelligence through the communication of symbols, the same methods of discrimination we rely on now might not hold up in the future. Reliable tests may prove harder and harder to find. For instance, realistic sounding chatbots may end up phoning people and holding functional conversations without being identified as automata. Future software may be able to generate images, audio, and videos of humans that look and sound like real humans but are actually fake, indistinguishable from "real" digital representations of people. Even though it seems that humans possess some intuition beyond formal logic, it might still become quite difficult to separate humans from machines in practice.

Can we design a test for intelligence that can be run in practice, though not passed by a machine? Intuitively it seems like we should be able to. But the missing ingredient here is proof. While it seems like we can set and pass tests which reliably draw a line between us and machines, we cannot prove it. We cannot say anything definitive at all about the relationship between computable functions and intelligence beyond the realm of the computable. We cannot bridge that gap in any meaningful way. There's nothing useful that can be said about intelligence beyond seeking to simulate it bit by bit in practice, by continually improving our machines and seeing what they are capable of.

This basic insight allows us to separate questions about machine intelligence that are useful from those that are not.

How resistant is language translation to simulation?—We can ask this question. Specifically, we can use the imitation game to quantify the difficulty of developing a machine that simulates human-level language translation to some level of accuracy. The potential availability of an answer renders the question meaningful.

Is human-level language translation beyond machines?—This question has no utility because it is not well-defined.

Can machines do language translation better than humans?—This question has no utility because it is not well-defined. Humans as a group provide the standard for recognizing what is linguistically correct. Doing machine translation well is about convincing other humans that the job is being well done; human opinion as a whole sets the standard.

Can machines do language translation better than the average bilingual human?—We can ask this question. As soon as restrictions are placed on human performance, machine simulation might be sufficient to surpass the ability of any given individual human at any given time. The question of who is doing a better job continues to be decided by humans as a whole, but assuming the judging process involves a larger group of humans, or a more skilled human, then it still makes sense to say that a machine can outperform an average human.

How soon will truck drivers by replaced by machines?—We can ask this question. Machines do not need to emulate truck drivers to replace them. Machines might well be better at driving than the average truck driver. They might also be cheaper. Nevertheless, humanity continues to define what counts as ideal truck driving. At the extreme limits, it becomes difficult to formalize how exactly a vehicle should drive, and the issue reverts back to human opinion. For example, the issue of who should AI kill in a driverless car crash resists logical formalization because it interacts with human life. According to Goodall (2014), autonomous vehicles will certainly crash, some crashes will certainly involve a moral component, and there is "no obvious way to encode complex human morals effectively in software".

Do chess-playing programs play better chess than any human?—Yes, they do. For instance, the Komodo chess engine can reach an Elo rating of higher than 3300, which is about 450 points higher than any human currently playing chess (Regan 2014).

Are machines now as good as humans at playing chess?—This question has no utility because it is not well-defined. Chess-playing algorithms continue to be strengthened by humans, implying that humanity knows more about chess than any given machine. The point at which further strengthening becomes impossible is undecidable.

Does the human brain hold less than 500 exabytes of information?—This question has no utility because it is not well-defined. We don't have any means to formalize the representation of human abilities and thus we don't have any means to decide whether a not a given system which uses under 500 exabytes can emulate the behaviour of the human brain. Any question which presupposes a complete representation of the human mind in its entirety is a useless question.

Here we can see a pattern: the questions that seek to benchmark machine intelligence against human intelligence are useless, while the questions that consider how useful machines can be to humans are useful. Accordingly, we recommend that frameworks for evaluating the quality of machine "simulation" should be focused, not on the mimicry of human thinking, but on utility.

7. *Conclusion*

In recent years there have been suggestions of a possible future technological singularity, at which point computer programs would begin improving themselves recursively (e.g. Hutter 2012; Schmidhuber 2012). This concept, first identified by von Neumann (Stanislaw 1958), refers to the point at which machines start designing machines better than themselves, leading to a runaway effect, or intelligence explosion. According to this vision, smart machines will start designing successive generations of increasingly powerful minds, creating intelligence that far exceeds human intellectual capacity or control. Proponents perennially see the singularity as being 15 to 20 years off (e.g. Kurzweil 2005).

However, this concept is based on the flawed perspective of machines reaching a point where they emulate human intelligence. As we have seen, it is not possible to define or identify such a point. Given that the question of emulation is useless, then the idea of human labour being superseded is also meaningless. The rise of machine intelligence will not eliminate the value of human labour, but rather shift it away from repetitive formal tasks towards more complex psycho-social activities that are not as easily automated (see Turing 1951).

The focus in philosophy on directly contrasting human and machine intelligence has proved misguided. Over the past three decades a paradigm shift has occurred in the field of AI, with the focus moving away from a theory-driven quest to emulate the wholesale architecture of the mind, towards a data-driven approach which aims to achieve practical results in restricted domains (Cristianini 2014). A machine does not need to represent the full range of human abilities for it to be smart in some way. Human intelligence is social, embodied, and enactive, and very poorly described as symbol-processing or rule-following. AI, by contrast, aims to automate those aspects of human behaviour that are not unduly sophisticated.

AI is mostly developed and applied in limited domains, where computers' superior abilities in dealing with vast amounts of data quickly and following rules exactly are of greatest benefit. The performance of these systems is often already so far beyond human ability that comparing human and machine becomes wholly irrelevant. In other domains, poorer performance by machines will be tolerated as long as the "digital labour" is cheaper and more reliable. The question of who does the job better thus becomes moot.

Over the last 80 years, the process of computation defined by Church and Turing has proved extraordinarily useful to humans, and transformed modern society. In contrast, endless debates on whether the human mind can be matched or surpassed by AI are guaranteed to lead nowhere. Thus, the relevant questions for philosophy are not "is the mind a machine?" or "will there be a technological singularity", but rather, "how useful will machines be?" and "how will they change our lives?"

References

Aaronson, S. 2005. PHYS771 lecture 10.5: Penrose.

Aaronson, S. 2006. "Shtetl-optimized." The blog of Scott Aaronson. Reasons to believe.

Church, A. 1936. "An unsolvable problem of elementary number theory." *American journal of mathematics* 345–363.

Copeland, B. J. 2002. "The Church-Turing thesis." The Stanford Encyclopedia of Philosophy (Spring 2002 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/spr2002/entries/church-turing/>

Copeland, B. J. 2003. "The Turing Test." In *The Turing Test*. New York: Springer: 1–21.

Cristianini, N. 2014. "On the current paradigm in artificial intelligence." *AI Communications* 27 (1): 37–43.

Cristianini, N. 2016. "A different way of thinking." *New Scientist* 232 (3101): 39–43.

Cuffaro, M. E., and Fletcher, S. C. (eds.). 2018. *Physical perspectives on computation, computational perspectives on physics*. Cambridge: Cambridge University Press.

Earman, J., and Norton, J. D. 1993. "Forever is a day: Supertasks in Pitowsky and Malament-Hogarth spacetimes." *Philosophy of Science* 60 (1): 22–42.

Fortnow, L. 2009. "The status of the p versus np problem." *Communications of the ACM,* 52 (9): 78–86.

French, R. M. 2012. "Moving beyond the Turing test." *Communications of the ACM,* 55 (12): 74–77.

Goodall, N. J. 2014. "Ethical decision making during automated vehicle crashes." *Transportation Research Record* 2424 (1): 58–65.

Gold, E. M. 1967. "Language identification in the limit." *Information and control* 10 (5): 447–474.

Harnad, S. 1992. "The Turing test is not a trick: Turing indistinguishability is a scientific criterion." *ACM SIGART Bulletin* 3 (4): 9–10.

Hayes, P. and Ford, K. 1995. "Turing Test considered harmful." In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. San Francisco: Morgam Kaufman Publishers: 972–977.

Hodges, A. 2009. *Alan Turing and the Turing test*. New York: Springer.

Hofstadter, D. H. 1980. *Godel, Escher, Bach: An eternal golden braid*. New York: Penguin Books.

Hutter, M. 2012. "Can intelligence explode?" *Journal of Consciousness Studies* 19 (1–2): 143–166.

Kieu, T. D. 2004. "Hypercomputation with quantum adiabatic processes." *Theoretical Computer Science* 317 (1–3): 93–104.

Kurzweil, R. 2005. *The singularity is near: When humans transcend biology*. New York: Penguin.

Legg, S. and Hutter, M. 2007. "Universal intelligence: A definition of machine intelligence." *Minds and Machines* 17 (4): 391–444.

Levesque, H. J. 2014. "On our best behaviour." *Artificial Intelligence* 212: 27–35.

Levin, L. A. 2003. "The tale of one-way functions." *Problems of Information Transmission* 39 (1): 92–103.

Li, M. and Vitányi, P. 2008. *An introduction to Kolmogorov complexity and its applications*. New York: Springer.

Longo, G. and Paul, T. 2011. "The mathematics of computing between logic and physics." B. Cooper and A. Sorbi (eds.). *In Computability in Context: Computation and Logic in the Real World*. London: Imperial College Press: 243–273.

Maguire, P. and Maguire, R. 2018. "On the measurability of measurement standards." *Croatian Journal of Philosophy* 19 (3): 403–416.

Maguire, P., Moser, P., and Maguire, R. 2015. "A clarification on Turing's test and its implications for machine intelligence." *Proceedings of the 11th International Conference on Cognitive Science*: 318–323.

McDermott, D. 2015. "What was Alan Turing's imitation game? Assessing the theory behind the movie." *The Critique*, January. URL: http://www.thecritique.com/articles/what-was-alan-turings-imitation-game/.

Müller, V. C. and Bostrom, N. 2016. "Future progress in artificial intelligence: A survey of expert opinion." In V. C. Müller (ed.). *Fundamental issues of artificial intelligence*. New York: Springer: 553–570.

Odifreddi, P. 1992. *Classical recursion theory: The theory of functions and sets of natural numbers*. Amsterdam: Elsevier.

Penrose, R. 1994. *Shadows of the mind*. Oxford: Oxford University Press.

Post, E. L. 1936. "Finite combinatory processes-formulation 1." *The Journal of Symbolic Logic* 1 (3): 103–105.

Regan, K. 2014. "The new chess world champion. Godel's Lost Letter and P=NP." Dec 28. URL: https://rjlipton.wordpress.com/2014/12/28/the-new-chess-world-champion/.

Rice, H. G. 1953. "Classes of recursively enumerable sets and their decision problems." *Transactions of the American Mathematical Society* 74 (2): 358–366.

Sandberg, A. and Bostrom, N. 2011. "Machine intelligence survey." *FHI Technical Report* 1.

Schmidhuber, J. 2012. "Philosophers and futurists, catch up! Response to the Singularity." *Journal of Consciousness Studies* 19 (1–2): 173–182.

Searle, J. R. 1980. "Minds, brains, and programs." *Behavioral and brain sciences* 3 (3): 417–424.

Shannon, C. E. and McCarthy, J. 1956. *Automata studies*. Princeton: Princeton University Press.

Shieber, S. M. 1994. "Lessons from a restricted Turing Test." URL: arXiv preprint cmplg/9404002.

Shieber, S. M. 2007. "The Turing Test as interactive proof." *Nous* 41 (4): 686–713.

Silver, D. et al. 2017. "Mastering the game of Go without human knowledge." *Nature* 550 (7676): 354.

Sloman, A. 2002. "The irrelevance of Turing machines to artificial intelligence." In M. Scheutz (ed.). *Computationalism: New Directions*. Cambridge: MIT Press: 87–127.

Turing, A. M. 1936. "On computable numbers, with an application to the Entscheidungsproblem." *Journal of Mathematics* 58 (345–363): 5.

Turing, A. M. 1947. *Lecture on the Automatic Computing Engine*. In Turing and Copeland 2004.

Turing, A. M. 1948. *Intelligent machinery*. In Turing and Copeland 2004.

Turing, A. M. 1950a. "Computing machinery and intelligence." *Mind* 59 (236): 433–460.

Turing, A. M. 1950b. Programmers. Handbook for Manchester Electronic Computer, University of Manchester Computing Laboratory. A digital facsimile of the original may be viewed in The Turing Archive for the History of Computing document. http://www.AlanTuring.net/programmers handbook.

Turing, A. M. 1951. *Intelligent machinery, a heretical theory*. In Turing and Copeland 2004.

Turing, A. M. 1954. *Solvable and unsolvable problems*. In Turing and Copeland 2004.

Turing, A. M. and Copeland, B. J. 2004. *The essential Turing: seminal writings in computing, logic, philosophy, artificial intelligence, and artificial life, plus the secrets of Enigma*. Oxford: Clarendon Press Oxford.

Ulam, S. 1958. "John von Neumann 1903–1957." *Bulletin of the American mathematical society* 64 (3): 1–49.

Warwick, K. and Shah, H. 2015. "Can machines think? A report on Turing Test experiments at the Royal Society." *Journal of Experimental and Theoretical Artificial Intelligence* 28: 1–19.