Turing test does not work in theory but in practice

Pertti Saariluoma[1] **and Matthias Rauterberg**[2]

[1]Information Technology, University of Jyväskylä, Jyväskylä, Finland

[2]Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract - *The Turing test is considered one of the most important thought experiments in the history of AI. It is argued that the test shows how people think like computers, but is this actually true? In this paper, we discuss an entirely new perspective. Scientific languages have their foundational limitations, for example, in their power of expression. It is thus possible to discuss the limitations of formal concepts and theory languages. In order to represent real world phenomena in formal concepts, formal symbols must be given semantics and information contents; that is, they must be given an interpretation. They key argument is that it is not possible to express relevance in formal concepts only. Consequently, computational models can be valid in some specific interpretations, and the Turing test can therefore only work in specific tasks.*

Keywords: Turing test, autonomous technologies, design science, mind, machine, artificial intelligence

1 Introduction

The Turing test is probably the best-known thought experiment in Artificial Intelligence (AI) [13, 14]. Its goal is to answer one question: Can machines think? Thus, tests form an important component of the argument comparing human and machine information processing [5, 11, 19]. Since the discussion on intelligence of machines underpins much of the modern computational cognitive psychology and philosophy of the mind, it is also essential in developing robots and autonomous systems [2, 7,].

The Turing test is essential, as it enables computational thinking to be viewed through multiple lenses. The notion of computational thinking has many different forms but here, it refers to the use of computational concepts to investigate intellectual information processes [15]. Prime examples of such thinking are the Turing machine and modern computers [13, 14].

Computational thinking, that is, investigating information processes in different contexts by means of computational (algorithmic and mathematical) concepts, is a vital process today, as it has allowed us to realize such intellectual processes by means of computers, information systems, and robots. In a social context, this means that machines are used for new and more demanding tasks. Thus, the issue of linking machine and human intelligence has become central to scientific thinking.

Essentially, the Turing test is an imitation game. Like any good experiment, it has two conditions. In the case of control conditions, it is assumed that there is an opaque screen. On one side of the screen is an interrogator whose task is to ask questions and assess the nature of the answers. On the other side, there are two people, A and B. The task of A and B is to answer the questions, and the task of the interrogator is to guess who has given the answer. In the case of experimental conditions, B is replaced by a machine (computer) and again, the interrogator must decide whether it was the human or the machine who answered the questions. A teletypewriter is used to eliminate the problems caused by the quality of the voice communication [14].

The decisive criterion in these experiments is the capacity of the interrogator to say whether the answer was given by human or machine. If the interrogator cannot do this, then the machine has passed the test. Therefore, the outcome of the experiment is that machines can think, as they can perform human tasks in such a way that it is impossible for a competent observer to see the difference between human and machine.

It is obvious that the Turing test is a particularly clever idea, but in what way exactly? Generations of renowned researchers have considered all aspects of the test [1]. The test itself was an innovation intended to justify Turing's computational theory of mind, which was criticized even before the test [13, 14]. Perhaps one of the most outspoken of the pre-test critics was Turing's close colleague and friend Ludwig Wittgenstein [17, 18]. Of course, his critical views became public after the publication of the test, but as Turing and Wittgenstein had close contacts in Cambridge, their differences of opinions in respect to computational thinking had developed before the Second World War [8]. Even today, we must still ask the question: Does the mind compute, hyper-compute, or even more? [9]

Since the Turing test was published, it has been discussed extensively, and many researchers such have investigated it[4, 5, 11, 12]. This is unsurprising, as the test is still considered somewhat elusive. On one hand, it is easy to identify machines that can beat people in their particular fields, for example, chess machines and pocket calculators. On the other hand, however, no general man-like Leviathan exists.

Studying the division of opinions around computational thinking and the Turing test is important, as the clarity in this issue would enhance the conceptual clarity around all of computational thinking. In order to analyse the conceptual properties of Turing's test, it is first worth considering some of the major critical arguments against the idea that human thinking is realizable using machines or that it is possible to implement all human information processes by means of computational machines. We then need to look at the practical problems specific to computers' information processing to facilitate an understanding of any hidden conceptual issues that could shed light on the problems and criticisms relating to Turing's work. Finally, it is essential to ask what implications such criticisms have for the role of the Turing test in theoretical and practical computational thinking.

2 Critique of machine thinking

The idea of the human as a computing machine has raised a number of critical points, which are important in analysing the scope and limitations of computational thinking. Presumably, this is why Turing gave so much space to the main critiques of the time when presenting his thought experiment [14]. Some of the critical arguments, such as "ESP," "theological" and "head in the sand" objections, can be put aside as they do not have much relevance in modern science. ESP, for example, is an unclear phenomenon, and the consciousness argument can also be disregarded, as it is not relevant in investigating the ideas behind the Turing test. In fact, Turing never claimed that his test was a test of consciousness. However, there are many important points in this discussion that are worth considering here.

As noted above, Ludwig Wittgenzstein was apparently the first critic of Turing's thinking; fundamentally, he really disliked the idea of people being machines [16, 17, 18]. In the Bluen and Brown books, for example, he wrote, "The trouble is rather that the sentence, 'A machine thinks (perceives, wishes)' seems somehow nonsensical. It is as though we had asked 'Has the number 3 a color '.[18]

Wittgenstein put forward a number of points against people being computers, beginning with the idea that people feel pain and machines do not. One could call this a biological argument. However, the most relevant counterargument here is the argument concerning "seeing as," i.e., interpreting elementary symbols such as percepts and words. To Wittgenstein [16, 17, 18], anything we perceive can have a multitude of aspects or interpretations. Words, for example, impart their meaning as soon as they are used. For Turing, this problem was not an issue, because he regarded symbols as a given. In essence, numbers are well-defined symbols, but Wittgenstein critically asked whether words and images are like numbers, that is, well-defined objects, and recognized the problems of automatic encoding in his ostensible "language game." Consequently, he spent the last decades of his life analysing the process of giving meanings to symbols.

Wittgenstein's point was subsequently presented in an advanced form by Searle [12]. The latter points out that machines are not able to concentrate on essential aspects of chess positions; rather, unlike people, they scan all possible alternatives mechanically in order to find a solution. This, of course, is as true today as it was in the seventies. People are still as capable of concentrating on the essentials of chess as they are on giving meanings to words, patterns, and any other type of sensory input.

A somewhat different critical argument was presented by Lady Lovelace [14]. Her argument was focused on Babbage's analytical engine, which was one of the first versions of the thinking machine. Her claim was that such engines could not initiate anything, particularly anything new. Turing could not accept this argument, but neither did he present a very convincing counterargument.

Finally, another point worth looking at again is the "informality of behaviour argument," which was also considered by Turing [14]. The core of this argument is that there are no rules that can mechanically explain human behaviour. People may stop when they see a green light, but it may also be that they do not. Thus, human behaviour cannot be determined fully by a given set of rules in the way that machines are. Instead, people apply rules flexibly in their lives and can usually decide which rules are relevant in any given context.

The above arguments are sufficient for our purposes, considering the scope of this paper. Our next question is whether it is possible to ascribe any practical meaning to these points or whether they are, in fact, merely conceptual. Second, it is vital to consider why it is so difficult for machines to select between essential and inessential courses of actions.

3 Practical problems with machine thinking

Real computational systems also have difficulties with meaning and content. Problem-solving programs are a prime example of human thinking realized by machines. A classic example of this is the chess machine, which is programmed to beat human chess masters.

In implementing problem solvers, two fundamental problems must be addressed. First, computers searching huge problem trees have to deal with an issue known as exponential search, which seems not to trouble people. Exponential growth in a search means that when the depth of the search is increased, the size search tree grows very fast. Thus, chess-playing computers have to generate millions of moves to reach human levels of performance, since people generate some 50 moves per problem position [6]

A somewhat similar problem is pattern matching. While a CEO can easily see essential patterns in economic development, it is very hard for computers to develop a pattern-matching system to facilitate such strategic management. Indeed, in many cases, pattern matching is even difficult for people. A similar problem is conflict resolution, where there may be several possible patterns leading to potential action, but one has to identify the best choice.

These practical problems are connected to original critiques. If it were possible to identify the right aspects of patterns such as words or signs syntactically, pruning search trees would not be a problem. It would only be necessary to indicate which branches could not be generated. In the same way, pattern recognition problems are easy to solve, if only it were possible to solve the problem of aspects. Both of these problems arise from the fact that syntax does not provide sufficient information for semantics selection [12].

4 Relevance: The rebuttal of computational concepts in modelling the human mind

Scientific and other representational languages have their scope and limitations. For example, it is not possible to find a natural number that expresses such issues as the relationship between the side and the diagonal in a square or the value of PI. To represent such entities, it is essential to use real numbers. Similarly, in behaviourist psychology, it is not possible to consider the properties of human memory or mental images, as they are more than simply stimulus-response concepts. Finally, had we restricted ourselves to Dalton's concept of the atom, modern nanophysics would also have been impossible. In science, progress is always about finding new concepts and exceeding the limitations of old ones [3, 10].

The limitations in the basic concepts of a scientific approach can be characterized by *power of expression* [10]. In essence, power of expression describes the limits of a conceptual system and expresses what can be thought when a particular set of concepts is being used. It can be used very naturally in this case, as it enables us to questions the limits of computational concepts and therefore the real meaning of Turing's test.

Put simply, computational concepts are based on abstractions devised by Turing. When he postulated the Turing machine, Turing assumed that he was similar to a mathematician describing how to solve any mathematical problem. What the mathematician does is to manipulate symbols on squared paper following a given set of rules. For the sake of simplicity, Turing assumed that the machine had an infinite tape featuring zeros and ones. The task of a mathematician following the rules was to manipulate the numbers on the tape. The numbers were supposed to represent symbols, which could be numbers, but also Chinese symbols. Thus, like many others after him, Turing assumed that the Turing machine was, in some senses, a model of the human mind.

However, Turing did not specify how the real world symbols and their meanings were associated with the Turing model. The associations between the number combinations on the tape, the symbols, and the references to symbols are given but not processed. Thus, the most important action is omitted from the computational thinking process, which involves figuring out how computational models can be combined with reality.

In mathematical concepts and the metascience of mathematics, *relevance* refers to the rule determining which elements of any mathematical set (of elements or functions) belong to one category (relevant) and which to another (irrelevant). In terms of Turing machines, one should be able to say which combination of zeros and ones are relevant and which are not. However, this is impossible in mathematical or formal concepts, as the theory language does not have the power of expression typical to natural languages. Since Turing machines and mathematical models are constructed by means of abstracting semantic and thought content, one can no longer present what is relevant in some concrete context.

When the contents are abstracted, it is impossible to produce sense-making semantics; in other words, it is impossible to define what is true and what is false or what is right and what is wrong. This means that only interpretation in terms of real world concepts, i.e., programmed semantics, makes it possible for AI systems to have any relevance.

Turing was not the first person that encountered this problem of linking formal systems to reality at the start of the last century. Before him, Ludwig Wittgenstein) had seen the same problems. In his "Tractatus-Logico Philosophicus," he gave a logical explication for the problem of human experience limitations [16]. Obviously, he had noticed that logical (or syntactic) symbols were void of meaning, and this is why in his later philosophy, he adopted the process of giving meaning as his topic. Of course, this explains his criticisms of Turing's computational thinking theory; he had seen the very problem that Turing had simply brushed under the carpet.

In order to be able to model reality, one must have a representation that contains the correct information. Otherwise, the model cannot be true and will misrepresent reality. "Correct" means that the symbols in the model have the correct semantics, and this presupposes that the information represented is right.

In fact, this strong AI premise is often defined to mean that the system has the right output when it has the right input. Or course, "right" in the given definition means precisely same as having the correct representational contents, as discussed above. "Meaning giving" thus becomes the most important problem. However, as Searle pointed out, syntax cannot generate semantics [12]. This is why the origins of meaning giving must be sought in human conceptualization and judgment processes, and this is why they are outside the framework of computational modeling.

The core explanation for the limitations in computational thinking is in the very abstraction process that creates symbolic information. The abstraction sets aside semantics and information contents. This is why syntactic models cannot represent a concrete state of affairs without interpretations. Proposition 3+5 = 4 is true, but that does not aid us in marketing fruit in a marketplace unless we know whether the formula describes apples, pears, bananas, or money. Similarly, it is valid to infer that "Napoleon was the Emperor of France" from the true fact that "the moon is not cheese," and "if the moon is not cheese, Napoleon was the Emperor of France." Whether the inference makes any sense is another issue. It is impossible to illustrate any relevant connection between Napoleon's role as Emperor of France and the fact that the moon is cheese.

Thus, the core issue seems to lie in the notion of relevance. Unless one can show that the semantic and information contents of a Turing model are relevant, the system does not work. Formal languages are void of content, and it is therefore impossible to determine their relevance in computational concepts. Computational concepts do not have the power to express relevance and for that reason, one needs an additional language and a scientific process to determine the relevancies and to implement systems in the real world. This, of course, works often in the modern world.

When the contents are abstracted, it is impossible to achieve sense-making semantics. It is also impossible to define what is true and what is false or what is right and what is wrong. This means that only interpretation in terms of real world concepts, namely, programmed semantics, makes it possible for AI systems to have any relevance.

However, in terms of the Turing test, the relevance requirement is devastating. It shows that machines can perform as well as people in processing information, but it does not show that people are computers. Obviously, people have capacities far beyond computational languages, and the failure of the Turing test as proof that a machine can function as a human does not influence the practical aspect of computational modelling.

5 Long live AI

Modern artificial intelligence is continuously achieving more interesting practical results. Autonomous systems, which broadly speaking, can redefine their goals during operation, are a good example of the capacity of emerging technologies. However, a merely intuitive interpretation of computational systems cannot make sense. It is time to go beyond the limits of computational concepts and admit that we need a new way of thinking that will incorporate computational representations of information contents.

In sum, this line of argument shows us that Turing's test is insufficient as proof of the identity of man as a machine. However, it can still play a very important role as the testing logic for computational models [5]. One must take the human mind as a criterion of how well machines have to perform to succeed in their main function, which is to replace human intellectual work. Only if a system passes the Turing test is it possible to view it as having a practical application. Thus, although the Turing test is theoretically dead, it certainly has a future in terms of designing new technical applications.

The important of Turing's logic is actually growing as technologies are improving. Autonomous systems, which are partly able to define their own goals, are a good example of what is in store in the future. Autonomous systems such as autonomous cars or flying devices are capable of changing their concrete goals depending on the situation. In general, autonomous systems form one of the key future developments in technology.

The core social importance of autonomous systems is rooted in their capacity to replace people in tasks that were traditionally carried out by people. In the case of many such tasks, it has never been possible to realize them technically. Typically, in those tasks, the system—the people and the machine—has to redefine ill-defined goals or redefine decision spaces.

Identifying solutions to such tasks is of central importance in future artificial intelligence. The classic example, as noted above, is the chess-playing machine. After not working well for five decades, it was suddenly possible for the machine to beat world champion chess player, Kasparow. This proves that there is a machine capable of performing an intellectual task as well as a human being. Of course, this was a machine with a special purpose; however, if chess were economically relevant, Deep Blue could have replaced all chess players, since it can be duplicated in millions.

The Turing test is vital for the technical realization of new intellectual tasks [20]. It also offers logic in assessing the performance of autonomous and other technical systems, and in designing ways to replace human work and leave new types of tasks to humans. The Turing test is essentially a test of performance rather than a test of how information is processed. Deep Blue, for example, has very little in common with human chess players in the way it processes information. However, commonalities are not essential from a technical or economic point of view. What is essential is that something can be achieved, and from this perspective, the Turing test is an excellent conceptual tool for designers. From ontological point of view, it is all too behaviouristic to be a test for can machines think like people. Therefore we can dinstinguish between *structural* equivalence and *functional* equivalence. The Turing test can help to identify functional equivalence but not structural equivalence.

6 References

[1] Tarek R. Besold. "Turing Revisited: A Cognitively-Inspired Decomposition. In: V.C. Mueller (ed.) *Philosophy and Theory of Artificial Intelligence* – SAPERE 5 (Studies in Applied Philosophy, Epistemology and Rational Ethics; Vol. 5, pp 121-132). Berlin: Springer. 2013.

[2] Margaret Boden. "Artificial intelligence and natural man". MIT Press. 1987.

[3] Nick Bostrom. "A history of transhumanist thought." *Journal of Evolution and Technology*, 14, (pp. 1-25). 2005.

[4] Mario Bunge. "Scientific research". Springer Verlag, 1967.

[5] Luciano Floridi., Taddeo, M., & Turilli, M. "Turing's Imitation Game: Still an Impossible Challenge for All Machines and Some Judges—An Evaluation of the 2008 Loebner Contest". *Minds and Machines*, 19, 145-150. 2009.

[6] Richard French. "Dusting off Turing test". *Science*, 336, (pp. 164-165), 2012.

[7] Adriaan de Groot. “Thought and choice in ch*ess”*. Mounton. 1978.

[8] Stevan Harnad, & Scherzer, P. “First, scale up to the robotic Turing test, then worry about feeling”. *Artificial Intelligence in Medicine*, 44, (pp. 83-89). 2008.

[9] Ray Monk. “Ludwig Wittgenstein. The Duty of Genius”. Jonathan Cape. 2009.

[10] Matthias Rauterberg. “Hypercomputation, uncounsciousness and entertainment technology”. In: Panos Markopoulos (Ed.), Fun and Games, LCNS 5294. (pp 11-20). Berlin: Springer Verlag. 2008.

[11] Pertti Saariluoma. “Foudational analysis. Presuppositions in experimental psychology. Routledge. 1997.

[12] Ayse Saygin, Cicekli & Varoll Akman. “Turing test – 50 years later”. *Minds and Machines*,10, (pp. 463-518). 2000.

[13] John Searle. “Minds, brains, and programs”. *Behavioral and Brain Sciences,* 3, (pp. 417-457). 1980.

[14] Alan Turing. “On computable numbers, with an application to the Entscheidungsproblem”, *Proceedings of the London Mathematical Society*, 42, (pp. 230-265). 1936.

[15] Alan Turing. “Computer machinery and intelligence”. *Mind*, 59, (pp. 433-460). 1950.

[16] Jeanette Wing. “Computational thinking”. *ACM Communications*, 49, (pp. 33-35). 2006.

[17] Ludwig Wittgenstein. “Tractatus logico-philosophicus”. Routledge, 1921/1974.

[18] Ludwig Wittgenstein. “Philosophical Iinvestigations”. Basil Blackwell. 1958.

[19] Ludwig Wittgenstein. “Blue and Brown Books”. Basil Blackwell, 1958.

[20] Jia You. “Beyond the Turing Test”. *Science*, 347, (pp. 116-116). 2015.