Revisiting the Turing Test: Humans, Machines, and Phraseology

Juliet Floyd, jfloyd@bu.edu

BISAN Lecture, February 6, 2023 - Scientists for Palestine

Forthcoming:
In *Nudging Choices Through Media – Ethical and Philosophical Implications for Humanity*, eds. James E. Katz, K. Shiepers and J. Floyd (Springer Nature Switzerland).

BOSTON UNIVERSITY

Table of Contents

Main Claims

1. Turing's "Imitation Game" (1950), the "Turing Test", was intended to be a social experiment in the evolution of human-to-human "phraseology" in the presence of machines: a language-game, in Wittgenstein's sense. It therefore remains relevant today (Chat GPT, GPT 3, Dall-e, etc.)

2. Turing regarded "intelligence" as a response-dependent, *emotional*, and hence social concept, like "freedom" or "agency" (Turing 1948).

3. For Turing it is human-to-human interaction in the presence of computational technology that matters, not merely human-machine interaction (Turing 1948).

Claims I Reject

1. The Turing Test answers the question of whether an individual consciousness or sentience might be had by a machine.
2. The Turing Test's primary point is epistemological: can machines fool humans by masquerading as human beings?
3. Turing was a computational mechanist and/or a functionalist about the human mind.
4. Wittgenstein and Turing were mutually "alien" to one another: on opposite sides of a philosophical dichotomy between methods of ordinary language and methods of formal logic (Monk).

My Argument

- Wittgenstein and Turing shared a matrix of foundational philosophical ideas about the nature of logic.
- They also discussed the nature, limits, and foundations of logic over many years.
- They drew from one another, as they both recognized.
- We have here a philosophical confluence of ideas forged over many years, not a conflict.

Wittgenstein $\rightarrow$ Turing

- Turing focussed on taking what we *say* and *do* with words seriously, and on the *limits* of formal methods, not only their power.
- Everyday language, including our "typings" of objects as they occur naturally in science and everyday life, are an evolving framework or technology. Turing stressed everyday human conversation and "phraseology" as foundational, the social and cultural dimensions of computations and algorithms.

Turing and Philosophy: Undergraduate 1931-4, Cambridge University

1. March 1933 (age 21) Turing read Russell's *Introduction to Mathematical Philosophy*, encountering the notion of philosophical "phraseology" and Wittgenstein's idea that logic is tautological. He began studying logic seriously.
2. Fall 1933-1935 Wittgenstein cancelled his "Philosophy for Mathematicians", something unusual: the group was too large. He dictated *The Blue and Brown Books* to a small group of mathematics students instead.
3. Turing spoke to the Moral Sciences Club in December 1933, arguing that "the purely logistic view of mathematics" is "inadequate" because there are many different ways to interpret mathematics (i.e., different phraseologies).

1936, "On computable numbers, with an application to the *Entscheidungsproblem*"

Turing gives a fundamentally persuasive analysis of "computation" by inventing the idea of (what was soon called) a *Turing Machine*.

The human interface, the human context of a shareable command, is *demonstrated* to be fundamental to the nature of computation.

The argument is *not* the application of a pre-existing blueprint in the literature. How did Turing come up with it?

Turing 1936: "On computable numbers, with an application to the *Entscheidungsproblem*"

Turing's greatest paper turns on Wittgenstein's method of language-games (Floyd 2012).

Cambridge 1934-6 (Turing's Graduate Fellowship Years)

Wittgenstein's *Blue and Brown Books* (1933-35):

- Construed logic anthropologically, in terms of "language-games" in which humans operate "mechanically" with signs, step-by-step, following command-tables. The question of *general* training is raised.
- "Can a machine think?" is a *grammatical* question requiring careful investigation, it is not yet clear: we can e.g. say that there is "thinking in the hand', but this does not refute the idea of human consciousness as irreducible. Human beings act both "mechanically" (e.g., as workers), and "creatively".
- Investigation of the concept of "thinking" must attend to how concepts are used in everyday life. Grammatical "experiments" with specific, sometimes invented "phraseologies" – i.e., *comparison* of different language-games – is the method.

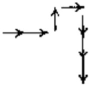
Wittgenstein's *Brown Book §41*:

a	→
b	←
c	↑
d	↓

Command table

"a a c a d d d"

Command



Result

Hilbert's *Entscheidungsproblem* (1928)

Turing solved the following problem:

Is there a definite method (algorithm) for deciding, Yes or No, whether or not a sentence of a theory couched in a first-order formal logical system follows from the axioms?

(Hardy (1929): Do mathematicians make discoveries by turning the handles of a miraculous machine?)

Turing's 1936 answer to Hilbert: No.

1. There exists *no* general algorithm (no "definite method") to decide whether or not one sentence follows from another in a logic. "Thinking" is not in general reducible to algorithms.
2. More generally: there is no general algorithm to determine in advance what an algorithm will do on a particular input: the idea of working out the consequences of a concept or set of equations cannot be in general foreseen (Wolfram, *A New Kind of Science*: computational irreducibility).
3. Turing (1954): This shows that there will always be a need for humans to use *common sense* and novel modes of "phraseology".

Turing's Analysis of a "Definite Method": A Basic Logical Point

To obtain a positive result that there *is* a machine or algorithm or "definite method" that can accomplish a task, you must write down the method or algorithm, make the machine, or at least prove that it exists.

But to obtain a *negative* result, that something is impossible, you must give a persuasive *general characterization* of what doing that something would have looked like.

What *is* a "Definite Method"?

To satisfactorily resolve the *Entscheidungsproblem* one must:

- Analyze what is meant *in general* by a "formal system" and a "step" in a formal system in the relevant Hilbertian sense.
- This could not be done by simply writing down just another formal system or by discussing in the metalanguage various kinds of different formal systems.
- This is why the ("logic-free" versions of) λ-definability and the Herbrand-Gödel-Kleene equational systems were used, and also why Turing devised his machines with command-tables.

Negative Resolution of the *Entscheidungsproblem*, Co-extensiveness

- Church, Kleene and Rosser (1935) showed that the class of functions calculable in the Herbrand-Gödel-Kleene equational calculus is co-extensive with the class of λ-definable functions.
- Church (1935-36), building on Gödel (1931), demonstrated that there is no "effectively calculable" function which decides whether two λ-definable expressions are equivalent.
- Turing (1936) showed that no "machine" can "compute" the desired general procedure as an application of his wholly novel analysis. Appendix: the functions his "machines" can "compute" are just those that are λ-definable.

Kleene 1981a, 1981b

Turing [was] aiming directly at the goal....

Turing's computability is intrinsically persuasive in the sense that the ideas embodied in it directly support the thesis that the functions encompassed are all for which there are algorithms; λ-definability is not intrinsically persuasive ... and general recursiveness scarcely so (its author Gödel being at the time not at all persuaded)...

Wittgenstein's and Turing's point

To say what a formal system of logic *is*, we cannot just write down another formal system.

We have to get at this conceptually, philosophically.

We must clarify what a system of logic (or "definite method") *is* by looking at what it is used *for*, and make this *plain*.

Their answer:
Logic is used by human beings who speak, act, calculate and converse in an embodied social world (words are constantly embedded in evolving forms of life).

Sieg (2006)

Most importantly in the given intellectual context [the move from arithmetically motivated calculations to general symbolic processes that underlie them] has to be carried out programmatically by *human beings*: the *Entscheidungsproblem* had to be solved by *us* in a mechanical way; it was the normative demand of radical intersubjectivity between humans that motivated the step from axiomatic to formal systems.

JF: *And* the next step, from formal systems to their uses by embodied human beings.

The Stored Program Computer Concept

Turing (1936) constructs the *Universal Machine* that does the work of all machines, including itself. It can operate on its *own* commands.

- One cannot diagonalize out of the class of computable functions (some are partial): the Universal Machine gives us a robust "absolute" parameter for what counts as a *step* in a computation, not dependent upon any particular language or system of logic (Gödel: "a miracle").
- Turing's Universal Machine is responsible for the *ubiquity* of computational processing in our world, and its indefinite extent of application and ability to compress what is definable.
- The Universal Machine shows that there are no ultimate, general dichotomies between hardware, data, and software.

Wittgenstein, RPP I §1096 (MS 135, 1947)

Turing's "Machines". These machines are *humans* who calculate. And one might express what he says also in the form of games. And the interesting games would be such as brought one *via* certain rules to nonsensical instructions... One has received the order "Go on in the same way" when this makes no sense, say because one has got into a circle. For that order makes sense only in certain positions.

(Wittgenstein knows that Turing's proof is philosophically general: it turns on constructing a tautological machine that is circular, ending in nonsense (not the Halting Argument, but "Do What You Do") (Floyd 2012).)

Turing on Human-Machine Interaction: Lecture to the London Mathematical Society (1947)

The Masters [i.e., mathematicians] are liable to get replaced because as soon as any technique becomes at all stereotyped it becomes possible to devise a system of instruction tables which will enable the electronic computer to do it for itself. It may happen however that the masters will refuse to do this. They may be unwilling to let their jobs be stolen from them in this way. In that case they would surround the whole of their work with mystery and make excuses, couched in well-chosen gibberish, whenever any dangerous suggestions were made. I think that a reaction of this kind is a very real danger.

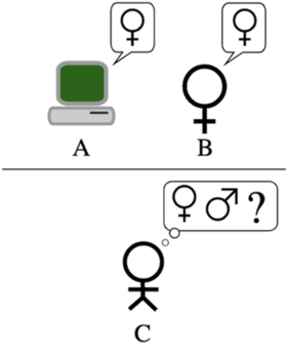
Origins of the Turing Test: Turing and Wittgenstein

- 1937 Discussions in the Cambridge Botanical Gardens.
- 1939 Turing, working at Bletchley, still teaches at Cambridge, attends Wittgenstein's lectures on the foundations of math.
- 1942 Turing writes about types and phraseology, thanking Wittgenstein ("Reform of Mathematical Notation").
- 1942-44 Turing makes his "Notes on Notations".
- 1948 The Manchester "Baby": 1st electronic stored-program computer; Turing notes that it can "surprise" him. 1st programming manual for the Mark II (1950). Strachey: 1st computer-generated music (BBC, "God Save the Queen").
- 1950 Turing publishes his "Imitation Game".

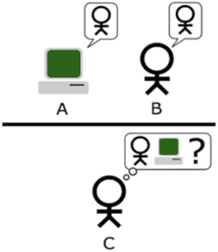
"Computing Machinery and Intelligence" (1950): The "Imitation Game"

- Turing publishes his article in the philosophical journal *Mind*.
- He reads out portions of it to his student Robin Gandy and giggles.
- Normal Malcolm writes to Wittgenstein to ask whether the article is a joke.
- Wittgenstein replies that he suspects it is "no leg-pull".

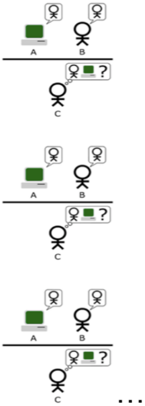
Turing 1950, the "Imitation Game": Gender as a Control



The Cartesian Turing Test for "Intelligent" Machinery



The Turing Test as a Social Experiment in Phraseology



Wittgenstein (1938)

What is called “winning” in chess might be losing in another game.

“What are we to say now?” –That is our theme.

What Turing is *Not* Doing with the Turing Test

- Trying to prove that machines *can* think.
- Assuming that behaviorism is true.
- Trying to prove that machines are conscious and capable of emotion.
- Trying to explain or deny the fact of consciousness.
- Trying to prove that humans are machines.
- Trying to prove that machines are indistinguishable from humans.
- Merely stipulating an operational or behavioristic definition of "intelligence".
- Assuming that disinterpreted operations with signs are capable of grounding meaning.

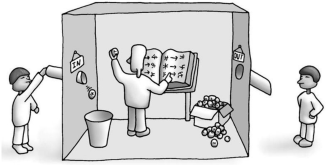
What Turing *Is* Doing with the Turing Test

- Showing that one cannot prove a negative result – e.g., that machines *cannot* think – because as yet one does not have a clear enough concept of "thought".
- Showing us how we might explore together the "emotional" effects of computational machinery on our ways of expressing ourselves, the *possibilities* for concepts and forms of life.
- Framing a repeatable, social, philosophically-minded human-to-human experiment in phraseology, or ordinary language: eliciting from us our working criteria, concepts, in actual, evolving forms of life *when we are unsure what to say*.

Turing 1950: Arguments Considered

1. The theological objection (souls, immortality).
2. The heads in the sand objection (too dreadful).
3. The Mathematical objection (human minds escape undecideability).
4. The Argument from Consciousness.
5. Arguments from the disabilities of machines.
6. Lady Lovelace's objection: Machines are not creative or surprising (answer: computational irreducibility).
7. Arguments from the continuity of the human nervous system (answer: randomness in discrete machines approximates this).
8. Arguments from the unpredictability of human behavior.
9. The argument from extrasensory perception.

Searle's Chinese Room



The Turing Test in Popular Culture

Etc. etc.....

Turing: “The Reform of Mathematical Notation” (1942-44)

The statement of the type principle given below was suggested by lectures of Wittgenstein, but its shortcomings should not be laid at his door.

Turing, "The Reform of Mathematical Notation" (1942-44)

Symbolic logic is a very alarming mouthful for most mathematicians, and the logicians are not very much interested in making it more palatable. It seems however that symbolic logic has a number of small lessons for the mathematician which may be taught without it being necessary for him to learn very much of symbolic logic.

In particular it seems that symbolic logic will help the mathematicians to improve their notation and phraseology.

Turing: "The Reform of Mathematical Notation" (1942-44)

We should conduct an extensive examination of current mathematical, physical and engineering books and papers with a view toward listing all commonly used forms of notation and examine them to see what they really mean. This will usually involve statements of various implicit understandings as between writer and reader. But the laying down of a code of minimum requirements for possible notations should be exceedingly mild, avoiding the straightjacket of a logical notation.

Turing: The Reform of Mathematical Notation" (1942-44)

It would not be advisable to let the reform [of notation] take the form of a cast-iron logical system into which all the mathematics of the future are to be expressed. No democratic mathematical community would stand for such an idea, nor would it be desirable.

Turing, "Intelligent Machinery" (1948): The "Intellectual" Search

We might arrange to take all possible arrangements of choices in order, and go on until the machine proved a theorem which, by its form, could be verified to give a solution of the problem ... Further research into intelligence of machinery will probably be very greatly concerned with "searches" of this kind. We may ... call such searches "intellectual searches".

Turing (1948): The Evolutionary Search

It may be of interest to mention two other kinds of search in this connection. There is the genetical or evolutionary search by which a combination of genes is looked for, the criterion being survival value. The remarkable success of this search confirms to some extent the idea that intellectual activity consists mainly of various kinds of search.

Turing (1948): The Cultural Search

The remaining form of search is what I should like to call the "Cultural Search'... [T]he isolated man does not develop any intellectual power. It is necessary for him to be immersed in an environment of other men, whose techniques he absorbs during the first 20 years of his life. He may then perhaps do a little research of his own and make a very few discoveries which are passed on to other men. From this point of view the search for new techniques must be regarded as carried out by the human community as a whole, rather than by individuals.

Thank You!