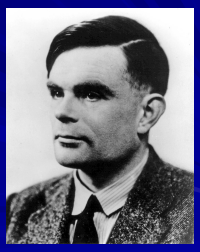
Turing to *ELIZA*

Turing to ELIZA

Online UNIQ Summer School, July 2020

Professor Peter Millican
Faculties of Philosophy and Computer Science
Hertford College, Oxford



1

1. Turing's Most Important Invention:

The Turing Machine (1936)

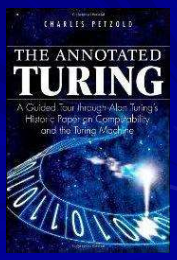
2

2

Textbook for Undergraduate Course

"Turing on Computability and Intelligence"
(Computer Science & Philosophy, Year 1)

- Charles Petzold, *The Annotated Turing*, contains Turing's classic 1936 paper in which he invented the Turing machine, the first theoretical model of an all-purpose digital computer.
- It also gives extensive explanation and discussion of the background and the paper.



3

3

"On Computable Numbers, with an Application to the *Entscheidungsproblem*" (1936)



- In 1928 the great German mathematician, David Hilbert, had highlighted the "decision problem" of mathematics: to find an"effective method" for computing whether any given mathematical proposition is provable or not.
- An "effective method" is supposed to be one that is clear, systematic, unambiguous, and specific.
- But these terms are all rather intuitive. To address Hilbert's problem, we need to understand more *precisely* what counts as "an effective method", and this is the role of Turing's "computing machines".

4

4

What is "effective computability"?

- An "effectively computable" procedure is supposed to be one that:
  - can be performed by systematic application of clearly specified rules,
  - without requiring any inspirational leaps or spontaneous intellectual insights.
- So to discover the limits of "effective computability:
  - we need to devise a way of encompassing all possible *mechanical* methods of inference: to design a *computing machine*.

5

5

"On Computable Numbers …"

- Turing focuses initially on the idea of *computing a "binimal" number* – an infinite sequence of binary digits ("0"s and "1"s).
  - He shows that some such sequences can be *defined* but not *effectively computed*: so there are limits to effective computability.
  - If you are interested in understanding how this issue plays out in his 1936 paper, and led to his resolution of the *Entscheidungsproblem*, see my paper "Turing and Machine-Like Intelligence" (2020, at http://www.millican.org/research.htm).

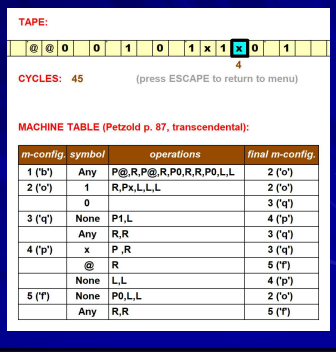
6

6

Turing to *ELIZA*

A Turing Machine in Action

Here we see a *Turtle System* Turing Machine simulator, at 45 cycles through the irrational (probably transcendental) number program that Turing gives in § 3 of his paper (*Petzold*, pp. 85-90).



m-config.	symbol	operations	final m-config.
1 ('b')	Any	P@,R,P@,R,P0,R,R,P0,L,L	2 ('o')
2 ('o')	1	R,Px,L,L,L	2 ('o')
	0		3 ('q')
3 ('q')	None	P1,L	4 ('p')
	Any	R,R	3 ('q')
4 ('p')	x	P ,R	3 ('q')
	@	R	5 ('f')
	None	L,L	4 ('p')
5 ('f')	None	P0,L,L	2 ('o')
	Any	R,R	5 ('f')

7

7

- The Turing machine consists of:
  - a long tape divided into squares, on which symbols can be written and later erased,
  - a read/write head (as in a tape recorder) which can write (or erase) symbols and also read them.
  - a "state", taking any of a specified finite range of values – this is the machine's active memory. Computation starts from the first, *beginning*, state.
  - instructions in the form of a "machine table", which tell it what to do in each possible circumstance:
- At each stage, depending on *the symbol read from the current square*, and *the current state*, it can:
  - Erase the symbol, or write a new one;
  - Move left or right on the tape;
  - Change to a new state.

8

An Amazing Result!

- A Turing machine can be made to calculate *anything that we know how to calculate*!
  - By designing an appropriate machine table, and a way of encoding relevant data into symbols on the tape, a Turing Machine can calculate anything that is calculable by even a modern computer armed with a sophisticated programming language.
  - So if something can be proved to be impossible for any Turing Machine (by exploiting its very limited range of operations), then we can conclude that it would also be impossible for any other computer.
    - And indeed impossible for any other *effective method* at all, according to what is known as the Church-Turing thesis.

9

2. The Halting Problem is Unsolvable

10

10

Refuting Decidability

- In his 1936 paper, Turing used his machines to specify a "computable number" (which he implicitly understood as a *binimal* fraction between 0 and 1, e.g. 0.00101101110 …), generating an infinite sequence of binary digits.
- But having defined these machines, Turing proved that certain of their properties – *in particular, whether they would ever get "stuck" and stop generating binary digits* – were undecidable. Hence decidability of effective computation (the *Entscheidungsproblem*) is unachievable.
- The impossibility of deciding whether a computing process will "get stuck" is quite general, and thus applies equally to standard programming systems. The unsolvability of this "Halting Problem" can be shown with remarkable ease …

11

11

The Halting Problem

- Suppose we have the text of a computer program *P* (in a standard programming language), taking input from a text file, *T*.
- We would like to have a program *H* which will examine *P* and *T*, and then reliably work out the answer to this question:
  - *Will program P, when run with input T, eventually halt, or will it never terminate?*
- *It follows from Turing's work that H is impossible*. This can indeed be proved, by showing that if it *were* possible, a contradiction would result.

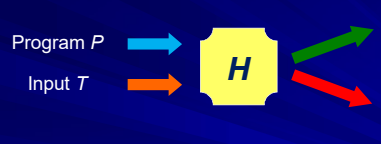
12

12

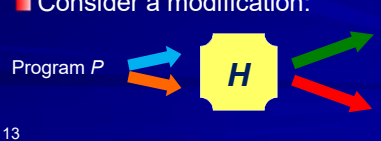
Turing to *ELIZA*

Proof by Contradiction

- Suppose we have such a program *H*.



- Consider a modification:



13

13

- Clearly if the initial program is possible, then the modified program must be possible too, because its only difference is that instead of taking *both* a program file *P and* a text file *T* as input, it uses the program file in place of *T* (hence its verdict is based on whether or not the specified program file *P* halts when given its own text as input).
- Having modified the *input* to our initial program, we now make an equally straightforward modification to its *output*. Here we assume that the output is generated by means of a simple "print" statement, which we modify in one case so that it goes into an infinite loop …

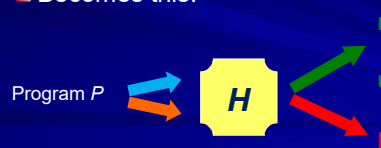
14

14

- We now modify our program so that this:



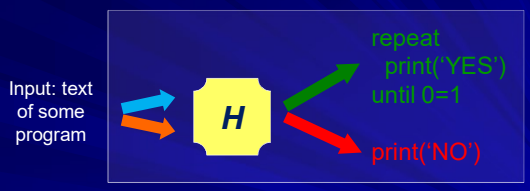
- Becomes this:



15

15

The Contradictory Program *Q*



- Does program *Q* halt when given itself as input? If it does, then it doesn't, and if it doesn't, then it does! So *Q* is an impossible program, and therefore *H* must have been impossible too.

16

16

3. Turing's Most Controversial Invention:

The Turing Test (1950)

17

17

Turing's 1950 Paper: "Computing Machinery and Intelligence"

- Published in the top Philosophy journal *Mind* (pp. 433-60), and one of the most cited papers ever!
- Introduced the *Imitation Game* (what we now call *the Turing test*) as a test for whether a machine can "think" (or, equivalently for Turing, is "intelligent").
  - For discussion of the paper's significance, you might be interested to read my 2013 paper "The Philosophical Significance of the Turing Machine and the Turing Test" (along with the 2020 paper previously mentioned, both of which are at http://www.millican.org/research.htm).

18

18

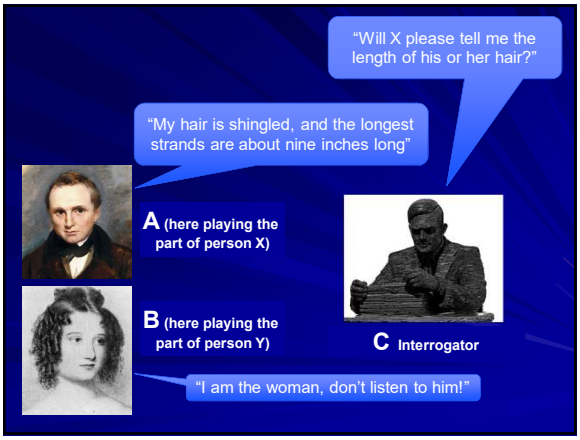
Turing to *ELIZA*

The Gendered Imitation Game

- Turing's introduces his "imitation game" in a form requiring an *interrogator* to guess the sex of two participants (whom he only knows as "X" and "Y", but knows to be of different sex), based on their answers to his questions. All questions and answers are communicated through written text.
- The man ("A") attempts by his answers to convince the interrogator ("C") that he is the woman – so he pretends to be female. The woman ("B") attempts to help the interrogator correctly identify her as the woman, by answering honestly. Introducing the game, Turing gives the following example:

19

19



20

The Computer Imitation Game

- The Turing Test arises when "a machine takes the place of *A* in this game". And the question of what will happen in these circumstances is supposed to "replace" the question "Can machines think?".
- Turing's paper is a bit confusing at this point: does he see the machine as trying to imitate a *woman*, or a *human*? And how exactly does he see success in imitation as connected to "thinking" or "intelligence"?
- Here we'll leave these issues aside, and interpret the Turing Test in the traditional manner (which fits, I believe, with what Turing himself intended):

21

21

The Turing Test
(As Traditionally Understood)



The computer "passes" the test if the Interrogator cannot reliably tell which is the computer and which is the human.

22

Turing Test Strengths and Weaknesses

- Can range over many different topics.
- Nicely "objective" in some ways:
  - Independent of appearance, voice etc.
  - Avoids issues of consciousness, feeling etc.
  - Ignores questions of the "soul" or "self".
- But seems chauvinistic in other ways:
  - Why is imitating a *human* so important? A computer might give itself away by being *more* intelligent than a human (e.g. at arithmetic)!
  - Why focus only on *verbal* behaviour? Animals show intelligence without any language.

23

A Sufficient, But Not Necessary Test?

- Turing himself notes the possible objection: "May not machines carry out something which ought to be described as thinking but which is very different from what a [human person] does?"
- He responds: "This objection is a very strong one, but at least we can say that, if, nevertheless, a machine can be constructed to play the imitation game satisfactorily, we need not be troubled by this objection."
  - He seems to be suggesting that *success* in the imitation game is *sufficient* to prove intelligence, but not a *necessary* condition. Thus he can acknowledge that an intelligent machine could "fail" the test by being *too* clever, or clever in a non-human way.

24

24

Turing to *ELIZA*

A Plausible Prediction, but …

- Turing famously predicts that by 2000:

  "… it will be possible to programme computers … to make them play the imitation game so well that an average interrogator will not have more than 70 per cent. chance of making the right identification after five minutes of questioning."

Whether or not this was actually achieved in practice, it looks quite plausible in retrospect. *If sustained and systematic research effort had been directed towards this goal*, it seems quite likely that it could have been achieved by 2000.

— But the prediction is plausible for a bad reason!

25

25

… an Unreliable Criterion

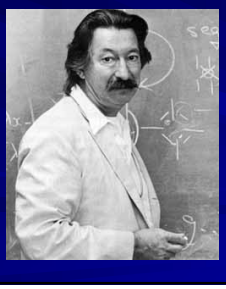
- Turing's way of presenting his prediction naturally suggests a criterion of progress towards machine intelligence, based on how well the program can fool an "average interrogator", and for how long.
- This has set the pattern for subsequent attempts at "passing the Turing Test" (as illustrated in the next slide), but this is highly regrettable, because

  as we shall see, it turns out that fooling an average interrogator is relatively easy to achieve, but not by techniques that plausibly involve genuinely intelligent information processing.

26

26

Debunking the Turing Test: Weizenbaum's ELIZA and Chatbots



TURING TEST SUCCESS MARKS MILESTONE IN COMPUTING HISTORY

Release Date 08 June 2014



An historic milestone in artificial intelligence set by Alan Turing - the father of modern computer science - has been achieved at an event organised by the University of Reading.

The 65 year-old iconic Turing Test was passed for the very first time by computer programme Eugene Goostman during Turing Test 2014 held at the renowned Royal Society in London on Saturday.

Eugene' simulates a 13 year old boy and was developed in Saint Petersburg, Russia. The development team includes Eugene's creator Vladimir Veselov, who was born in Russia and now lives in the United States, and Ukrainian born Eugene Demchenko who now lives in Russia.

The Turing Test is based on 20th century mathematician and code-breaker Turing's 1950 famous question and answer game, 'Can Machines Think?'. The experiment investigates whether people can detect if they are talking to machines or humans. The event is particularly poignant as it took place on the 60th anniversary of Turing's death, nearly six months after he was given a posthumous royal pardon.

If a computer is mistaken for a human more than 30% of the time during a series of five minute keyboard conversations it passes the test. No computer has ever achieved this, until now. Eugene managed to convince 33% of the human judges (30 judges took part - see more details below) that it was human.

27

27

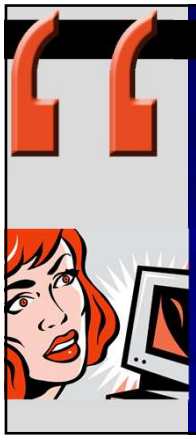
ELIZA the Psychotherapist

- In 1966, Joseph Weizenbaum published his *ELIZA* program and "DOCTOR" script, showing how very simple text manipulation could generate a plausible conversation.
  - The "DOCTOR" plays the role of a Rogerian psycho-therapist, largely echoing what the human says, expressing sympathy etc., and asking gentle questions to elicit their feelings.
  - Computer responses are generated by making small changes to the human inputs, exchanging first- and second-person etc.

28

28

ELIZA in action



- Men are all alike.
- **IN WHAT WAY?**
- They're always bugging us about something or other.
- **CAN YOU THINK OF A SPECIFIC EXAMPLE?**
- Well, my boyfriend made me come here.
- **YOUR BOYFRIEND MADE YOU COME HERE?**
- I need some help, that much seems certain.
- **WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP?**
- Perhaps I could learn to get along with my mother
- **TELL ME MORE ABOUT YOUR FAMILY.**

29

29



- My mother takes care of me.
- **WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?**
- My father.

…

- **WHAT ELSE COMES TO YOUR MIND WHEN YOU THINK ABOUT YOUR FATHER?**
- Bullies.
- **DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?**

30

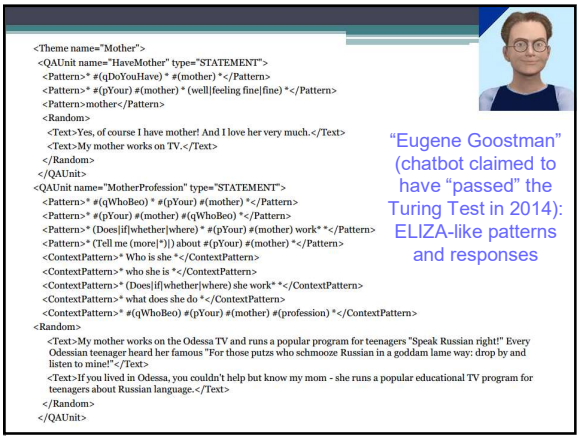
Turing to *ELIZA*

Simple Patterns and Responses

"… alike …"	"IN WHAT WAY?"
"something or other"	"FOR EXAMPLE?"
"My X"	"YOUR X?"
"I need Y"	"WHAT IF YOU GOT Y?"
"… my mother"	"… YOUR FAMILY …"
"My mother …"	"WHO ELSE …?"
<no matching pattern>	"DOES THAT HAVE ANYTHING TO DO WITH YOUR X?"

31

31



```
<Theme name="Mother">
 <QAUnit name="HaveMother" type="STATEMENT">
  <Pattern>* #(qDoYouHave) * #(mother) *</Pattern>
  <Pattern>* #(pYour) #(mother) * (well|feeling fine|fine) *</Pattern>
  <Pattern>mother</Pattern>
  <Random>
   <Text>Yes, of course I have mother! And I love her very much.</Text>
   <Text>My mother works on TV.</Text>
  </Random>
 </QAUnit>
<QAUnit name="MotherProfession" type="STATEMENT">
  <Pattern>* #(qWhoBeo) * #(pYour) #(mother) *</Pattern>
  <Pattern>* #(pYour) #(mother) #(qWhoBeo) *</Pattern>
  <Pattern>* (Does|if|whether|where) * #(pYour) #(mother) work* *</Pattern>
  <Pattern>* (Tell me (more|*)|) about #(pYour) #(mother) *</Pattern>
  <ContextPattern>* Who is she *</ContextPattern>
  <ContextPattern>* who she is *</ContextPattern>
  <ContextPattern>* (Does|if|whether|where) she work* *</ContextPattern>
  <ContextPattern>* what does she do *</ContextPattern>
  <ContextPattern>* #(qWhoBeo) #(pYour) #(mother) #(profession) *</ContextPattern>
<Random>
   <Text>My mother works on the Odessa TV and runs a popular program for teenagers "Speak Russian right!" Every
   Odessian teenager heard her famous "For those putzs who schmooze Russian in a goddam lame way: drop by and
   listen to mine!"</Text>
   <Text>If you lived in Odessa, you couldn't help but know my mom - she runs a popular educational TV program for
   teenagers about Russian language.</Text>
  </Random>
 </QAUnit>
```

"Eugene Goostman" (chatbot claimed to have "passed" the Turing Test in 2014): ELIZA-like patterns and responses

32

Making the Test More Difficult

- Any test that an ELIZA-style chatbot can pass clearly isn't a reliable test for intelligence!
  - So it's not enough just to fool "an average interrogator" 30% of the time in a 5-minute casual conversation.
- The test apparently needs to be made much more rigorous, by asking *specific*, *probing* questions that check for the ability to *understand*, in terms of *making links*, *drawing appropriate inferences* etc.
  - But the conceptual links we make in discussion often draw on all sorts of subtle features of our life and context (including human relationships). Could a computer realistically mimic these so as to be *indistinguishable*? And is such a demand sensible in a test for intelligence?

33

33

The Turing Test: Summary

- Turing acknowledged that passing his Test wasn't a *necessary* condition for intelligence.
- But unless interpreted fairly rigorously, it's not *sufficient* either, because it's too easy to pass due to *human* lack of critical discernment!
  - Much of our conversation is sloppy and careless;
  - Hence we too easily interpret sloppy and careless conversation as indicative of intelligence.
- And if interpreted rigorously, the test is *too* demanding to be useful: indistinguishability from human conversation isn't a sensible AI goal.

34

34

The *Tutoring* Test

- Perhaps a better form of test (suggested in Millican, 2013) is one focused on *revelation* of information processing – in a *teaching* task – rather than *deception*.
  - Enables focus on a specific domain, e.g. some aspect of chemistry, which is suitably deep with complex informational structures.
  - No need to pretend, and no expectation of indistinguishability: the test is how well the system can teach the theory (e.g can it teach as well as a human?).
  - Could provide *seriously useful* products, and also stimulate progress towards "intelligent" processing.

35

35

Uses for Chatbots

- Chatbots (or chatterbots) can be genuinely useful learning tools, one lesson being how quickly one discovers their limitations!
- But we can also benefit from our readiness to interpret their output as "intelligent" :
  - Most people prefer to interact conversationally.
  - Automated "help" systems can be useful even if their responses are pre-packaged.
  - Chatbot methods can often succeed in eliciting and identifying relevant information quickly.

36

36