Judging Chatbots Without Opening Them
Limitations of "black-box" tests.

Yet another comment relating to the (mythical) "Turing Test" for intelligence.

DRAFT BRAIN DUMP: Liable to change
Email comments and criticisms welcome.

Aaron Sloman
School of Computer Science, University of Birmingham.

Jump to Contents

Abstract (expanded 26 Aug 2014)

This is one of two documents reflecting on the (mythical) Turing Test. The other is concerned with my experience as a judge in the 2014 Turing Test event at the Royal Society of London, and the reasons why I think proposing a test for intelligence (which a careful reading of Turing's 1950 paper shows he did not do) is of little scientific or philosophical value, whereas a theory that can be shown to explain the competences of a wide variety of human-like individuals, including the various possible developmental trajectories, would be of great interest and importance. That document is http://www.cs.bham.ac.uk/research/projects/cogaff/misc/turing-test-2014.html

In contrast the take-home message of this document is that if a machine has infinite competence (in the sense proposed by Chomsky around 1965 -- for example, having a grasp of the grammar of a language that permits infinitely many sentences, or a grasp of some portion of mathematics that has infinitely many consequences, e.g. simple arithmetic), even if it has finite performance limitations (e.g. because of limitations of available memory, or energy, or resources for growth, etc.), then no finite set of discrete interactions with that machine can provide evidence that the machine's behaviour conforms to a particular theory, since, for any number N of observed interactions infinitely many different machines could share the same initial responses to those N inputs, while differing in subsequent interactions.

This is analogous to Karl Popper's argument (roughly) that no amount of supporting empirical evidence can raise above 0% the probability that a conjectured law of nature is true -- because any set of observations will be only an infinitesimal subset of the possible tests of the conjecture. However, the argument presented here does not depend on Popper's argument.

It follows that something more than observations of inputs and outputs is required to support a theory about what such a machine can do and how it does it. However, as in all deep science any such theory may remain subject to revision in the light of new evidence, unless the theory is based on detailed knowledge of how the machine was actually designed and built. We can't expect to have such knowledge about most animal brains, or the minds that they support, in the near future.

(Compare Rice's theorem, which makes an even stronger claim about what can be deduced when the machine's program is already known.)

NOTE:
Although I retired officially in 2002, the University of Birmingham School of Computer Science has continued to host my mainly philosophical research including making it very easy for me to construct rapidly changing web sites on philosophy of mind, philosophy of biology, philosophy of mathematics, philosophy of computation, and related topics. I hope this can help to counter widely held misconceptions of Computer Science as concerned only with solving practical problems and making useful machines. I see it as addressing deep scientific and philosophical problems, about a universe composed of matter, energy and information constantly interacting in changing ways.

This discussion note is
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/black-box-tests.html
A PDF version may be added later (but your browser should be able to produce one).

A partial index of discussion notes is in
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/AREADME.html

---

CONTENTS

Introduction
Judging Chatbots Without Opening Them
I.e. using "black-box" tests.

This is one of two documents arising out of the recent furore over the announcement that a chatbot had at last satisfied Alan Turing's prediction in 1950 (widely misunderstood and misrepresented as proposing a test for intelligence).

The first document (a) explains why I (perhaps foolishly) agreed to be one of the judges in the "Turing Test 2014" event, (b) explains why I think Turing did not propose a test for intelligence and (c) explains why attempts to improve on the test in the light of a large collections of criticisms of the test are misguided, since the very idea of a single behavioural test for intelligence is as flawed as a behavioural test for being a Turing machine. Testing is important for engineering purposes (though even there behavioural tests of a whole system can be seriously flawed as a method for finding faults). But when our aim is to answer scientific or philosophical questions, something much deeper than devising tests is required: namely producing an explanatory theory applicable to a wide variety of developmental trajectories in a wide variety of environments, as discussed in the other document: http://www.cs.bham.ac.uk/research/projects/cogaff/misc/turing-test-2014.html

This document goes into more technical discussion of why behavioural tests of computational systems in general are necessarily limited in what they can achieve, and in particular cannot be used to discover what sort of computational system a tested individual or machine is. This is related to Rice's theorem, a more general result familiar to theoretical computer scientists, mentioned briefly below. Ziegler (1974) may also be relevant, though I have not yet looked closely at it.

Turing is often wrongly reported as proposing a behavioural test for intelligence in his 1950 paper. Since some components of intelligence are computational (information processing) abilities, behavioural tests for intelligence should include behavioural tests for computational abilities. By showing the impossibility of using behavioural tests to establish exactly what computational powers a Turing machine has, and using the fact that TMs were designed by Turing to replicate certain sorts of human information processing abilities we can criticise ALL behavioural tests for intelligence. The key point is that input-output capabilities compatible with a particular Turing machine **T** revealed in any arbitrarily long, finite sequence of tests, could in principle be produced by infinitely many very different Turing machines **T'**, **T''**, **T'''**, ... or even by a machine that is much more limited than a Turing machine because it implements a fixed input output table and cannot produce any response for an input not in its table.

We can compare testing for intelligence with testing for whether something can compute and what it can compute, since a minimal requirement for intelligence is being able to process information, i.e. being able to compute, though the forms of information processing required by different species can vary enormously, both in their content and their complexity. For intelligent mammals and birds the types of intelligence produced by evolution still go beyond what AI systems and robots can do, except in very limited test environments, for which intensive training can 'program' a machine. In particular, no AI system or robot can match the bootstrapping of competence that members of many altricial species do between being helpless fledglings or neonates, and being expert nest builders, hunters, carers for offspring, socialisers, etc.

The impossibility of specifying a behavioural test for whether something can compute or what it can compute, can help to explain why it is impossible to specify a behavioural test for intelligence (or human intelligence).

Behaviourism, explicit or disguised in methodological reliance on experiments and statistics recording observed behaviours or responses to experimental probes, has seriously impeded progress in psychology.

Unfortunately, educational establishments teaching students psychology tend to overlook the fact that learning to build, test, criticise, and debug working models of mind (or models of various sorts of human and animal competences) is far more important for progress in a scientific psychology than learning to do statistical tests on data, usually given far more attention. (Clinical psychology has different aims from scientific psychology, and that may be the main cause of the distortions in teaching psychology).

Let's start with an apparently simple looking question that has hidden depths.

Jump to Contents

Can you decide whether a black box is a Turing Machine?

Suppose you have a Turing machine in a box that you cannot open, though you have a blank infinitely long tape on which you can write symbols to feed into the machine, and when you start the machine it will start reading the tape and go through a mixed sequence of reads, writes, moves left, moves right, and possibly halting, in accordance with its 'machine table', the list of instructions that define the machine's capabilities. You can restart with different contents on the state after each run of the machine.

You are told the finite set **S** of symbols the machine can possibly recognise on, or write to, locations on its tape. But you don't know its transition rules. You have to infer them from your tests.

The machine has a slot to feed in one end of a starting tape (infinitely long, of course), on which you can insert an arbitrary sequence of initial symbols using only symbols in S. It may either internally coil up previously read in portions of the tape inside itself, or allow the tape to project backwards out of the machine, opposite to the side from which a tape is read in.

If you prepare a test tape and press "start" the machine will do its stuff on the tape. It may or may not stop during your life time. If the machine eventually stops after a test tape is fed in you can then inspect the whole final tape.

While such a test is running you have no way of telling whether it will ever stop, since you don't know the machine's rules. You have no way of avoiding initial test sequences that will generate unending responses. That in itself makes finding out the machine's transition table by experimentation impossible. But there are also problems about what you can infer even if all the tests in your life time produce terminating behaviour. No matter how often that happens you cannot tell whether a new initial sequence that you have not yet already tried will produce non-terminating processes.

You can't even tell whether repeating one of your previous tests will sometimes give different final output tapes. If there's a random component in the machine then even after responding in the same way to a given input tape a million times it could do something different the next time.

Even without a random component, if it can keep count of the number of times it has been run, the computer could use the current counter value to alter answers to some questions put to it.

More subtly, if the machine is not a Turing machine with a fixed transition table, but can sometimes modify its table after dealing with a new tape, or if it can generate new sub-tapes that are always combined with future externally presented tapes, then in future if given an input tape that it has previously responded to, it may respond differently the second, and subsequent times. (The need for that in a model of animal information processing demonstrates that no fixed turing machine can model any organism that does significant learning of the sorts found in many animals.)

Comparison with the halting problem and other tests

This is related to the halting problem, but different from it. Turing proved in 1936 that no Turing machine can decide for any turing machine specification given to it whether that is the specification of a machine that will eventually halt if run.

However, our challenge is not the same as the halting problem challenge, for that is the challenge of defining a TM which, when given the detailed specification (i.e. machine table) of any TM (including itself), can tell by analysing the rules in the table, whether the specification will halt if run.

In our test the specification for the TM is not given, only the symbols it uses and its behaviour on inputs chosen by the tester. The task is not to decide whether it will stop on some input, but to use observed behaviour after a finite number of tests generated by feeding in starting tapes and recording the output after the machine stops to decide whether the black box contains a turing machine, and if so what sort of turing machine it is. E.g. is there some class of mathematical or logical problems that it can solve. The problem posed to the tester is to infer something about the machine table by producing a collection of input tapes and examining the output tapes. For example, the task may be to decide whether it computes the least common multiplier of two numbers, or whether it checks whether an inference in propositional or predicate calculus is valid, or to read in a specification for a board game and generate a program that can be run on the same machine, or some other machine, to play that game.

The claim I am making is that, because of the potential infinity of tests, if it is a turing machine, it follows that no matter which hypothesis appears to be most consistent with the results of some finite number of tests this does not rule out the existence of infinitely many different tests that have not been tried whose results, if tried, would be inconsistent with the hypothesis. So a very high success rate in a very large collection of tests of a hypothesis is of little significance in the long run, which, in this case is infinite.

It makes no difference whether our goal is to work out what information a human, or a TM, can gain about the competence of the machine from observations of its input-output behaviour. The answer is the same for both types of interrogator: neither can work out what's going on inside and derive with certainty, or even a high probability of correctness, either the precise machine table, or a general non-trivial categorisation of the machine table.

We could try narrowing down the search for diagnostic tests and consider whether it is possible, by running the machine with finitely many different input tapes, each representing specifications of a turing machine, to discover whether the black box contains a **Universal** Turing machine?

This cannot work, because no matter how many attempts you make to guess which method it uses to specify particular Turing machines (i.e. its "programming language" for specifying Turing machines) you cannot be sure you have found the right specification: even if you think you have discovered the machine's language and have conducted many tests and found that the tape always ends up as if the black box contained a particular sort of Universal Turing Machine, you cannot be sure that it will behave in accordance with your theory for all the infinitely many other possible tests you have not yet

tried.

If you are trying to find out whether the black box contains a Universal Turing Machine (UTM), that can emulate any other turing machine, you could assume that it contains a Turing machine and then use many test inputs to try to find out what sort of TM it contains, and including tests to find out whether it's a UTM, that has an input language for TM specifications, and for test tapes for each type of TM.

After many exploratory runs, using your best guess as to what its rules are, you may get as far as demonstrating that it properly runs a large number of test programs for a large collection of TMs. But no matter how many test runs fit your best theory, there remain infinitely many untried tests that can be encoded on its tape, and you cannot be sure that there is no crucial size of test that you have not tried such that for all input tapes above that size it will never behave in accordance with your current well supported theory.

In addition to all your tests completed by a certain time, if it is a UTM, then there will be infinitely many additional turing machines that can be programmed into it and infinitely many new tests for each of those machines.

You can try to be systematic and enumerate starting sequences, using the known set of symbols, by feeding in all the input sequences containing only 1 symbol, then all with 2 symbols, etc. etc. and for each sequence see whether the machine produces a new tape in accordance with your theory about its UTM.

But without knowing the machine's rules you have no way of avoiding test strings that will make the machine go into an infinite process, and if it goes into an infinite process you will have no way to tell that it is infinite. You'll just see the used part of the tape getting longer and longer, and never be able to tell that it will or will not stop if it has not yet stopped.

You are no better off if all your tests terminate. From the results of any number of tests that terminate, you will not be able work out how the machine will behave for all input sequences that you have not yet tried. So black-box computing machines are essentially inscrutable.

Jump to Contents

Huge Lookup Tables/Giant Lookup Tables/Humongous Lookup Tables

(Alternative headings here for search engines)

In particular, if you conjecture that you have guessed that the black box contains a particular machine table T, and have tested this by inserting millions of different input tapes and checked that the output is consistent with what output would be from T, you cannot thereby eliminate the possibility that the black box does not contain anything remotely like T, which could be a short set of rules. E.g. instead of T it might contain a very large but very simple specification that includes all the possible input sequences with length less than some very large number N, and all the corresponding output sequences, but with no rules for linking each input with its output other than their association in the table.

I.e. it could be a trivial type of computer that contains only a very large lookup table containing possible finite input tapes paired with possible output tapes, with mechanisms for comparing the actual input tape with the stored input tapes -- which might use something like an alphabetic ordering or more sophisticated techniques to speed up the search, or might not. If it finds a match then it spews out the corresponding stored output tape.

This is what used to be called a 'Humongous lookup table' or 'Huge lookup table' (HLT), or 'Giant lookup table' (GLT), in discussions of flaws of the Turing Test and in discussions of unconscious Zombies that behave like humans, a few decades ago.

You might think you could rule out the HLT hypothesis by watching the sequence of operations on the tape. If the machine reads in the tape erasing all the symbols as it goes, then goes back to the beginning and writes out the output, that might suggest use of the HLT. But if it does something much more complicated with many intermediate stages where sequences of symbols are written out and then replaced, before the final output is achieved, that might suggest that there is no HLT, but something more complex.

But the suggestion could be false, if after the machine uses the HLT to find the corresponding output tape it goes through an irrelevant but complex looking rigmarole designed to give the impression of computing something in a principled way, and then just writes out the previously found result. It would of course, need a lot of internal storage to hold the found output sequence while generating the smokescreen output, but a Turing machine table can be arbitrarily large (if we ignore the the finite amount of matter in the universe).

So you cannot guarantee to find out what the rules of the black box TM are from any finite number of completed tests, and you can't confirm any particular explanation of how the TM generates its results, merely by observing the input and output processes.

So you'll never be able to tell from behavioural tests whether a machine is a UTM, or what its machine table is if it's a non-universal TM. So, since Turing devised his machines by reflection on known capabilities of humans (at least human mathematicians) it follows that humans, whatever else they are, are capable of operating like Turing machines, though they all have memory limitations (what Chomsky referred to as performance limits, contrasted with competence limits). The above arguments show that inspection of input-output records cannot be used to determine whether something has human intelligence.

A corollary is that purely empirical psychology is an impossible dream and psychology needs to be supplemented with creative theorising, which we already knew was required in physics and chemistry. Unfortunately, a slavish empiricism, often based on misunderstanding Popper's demarcation between science and metaphysics has ruined the education of many psychologists, who learn how to evaluate experimental data but don't know how to build or evaluate deep explanatory theories. Piaget tried, but used inappropriate formal tools, including group theory, and propositional calculus. If he had learnt to program in a powerful AI programming language early in his career, the history of psychology might have been very different.

Probabilistic hypotheses fare no better.

Does increasing the number of successful predictions of the machine's responses to your tests increase the probability that your theory about its machine table is correct?

No, because, no matter how many runs you have been through collecting evidence, they are all finite input/output pairs, bounded in size by the largest inputs and outputs in your tests so far, and since the machine has an infinite tape, that means that you have sampled an infinitely small proportion of the total possible set of tests on the black box. The probability that you have the right theory based on the evidence you have collected will always remain at zero, if the black box contains a turing machine.

If you know something about the limits of the designer of the machine, and know that the designer is capable of producing only finitely many machines, each with a finite behavioural repertoire, then there will be some number N which is the maximum number of different behavioural histories of which machines produced by that designer are capable. In that case, if you assume that the machine in the box is M, and test the box by generating possible inputs for M, then the more different tests you do in which the black-box machine behaves as M would, then the smaller the proportion of remaining tests and therefore the smaller the chance of a remaining test producing behaviour that does not match your prediction.

But those are highly unrealistic assumptions because, as Noam Chomsky has often pointed out (unwittingly echoing Immanuel Kant?), humans have infinite generative potential in the things they understand even if that potential is limited by contingent factors such as limited brain size, or limited writing materials for external calculations. The same is true of any computer in which one can implement a recursive program for computing the factorial of I, an integer, as in:

```
define factorial(I);
    if I < 2 then 1
    else
        I * factorial(I-1)
    endif
enddefine;
```

A computer that 'understands' that definition, i.e. is capable of running commands like:

```
    print(factorial(99))
```

can in principle use it to compute the factorial of any number, no matter how large. Like humans, it has "infinite competence", even though its performance will always lie within fixed finite bounds, because of limitations of machine size and speed. (The point about humans having infinite competence with limited performance was first made explicitly by Noam Chomsky, in the 1950s or 1960s. I think it was previously understood by Kant and other philosophers of mathematics, but none of them expressed it as well as Chomsky.)

In practice a computer with infinite competence may be limited by the way it represents numbers (e.g. a maximum bit size), or by the available memory size, so that there is a maximum size of integer that it can handle. Machines may also be limited by the memory space available for recursion, which might cause the procedure to run out of space. Tail recursion optimisation is a technique sometimes used to overcome that limit. But that would not overcome the previous constraints.

So machines with finite limits in practice (performance limits) may have within them rules that are infinite in scope, and in such cases trying to infer the nature of the rules on the basis of observation of behaviour is doomed. A theory about the mechanisms will have to be discovered in another way. One way may be to try to understand previously unnoticed features of the evolutionary processes that created the mechanisms, as in the the Meta-Morphogenesis project.

This is related to Chomsky's ideas about human linguistic competences, but his ideas need to be supplemented with theories that put evolution, development and use of language in the context of many other competences, including some that require powerful **internal** languages for internal uses rather than communication with others.

Jump to Contents

[Digression: Karl Popper on interrogating the Universe

The previous comment seems to be closely related to Popper's argument that it is impossible for confirming evidence to increase the probability of truth of a universally quantified scientific theory, roughly because no matter how many observations have been made that support a theory they will always be an infinitesimally small subset of the possible observations. [REF] See also this note on the Popper-Miller theorem.

Popper's arguments about the inability of evidence to increase the probability of a universally quantified statement are related to the argument here that no amount of behavioural evidence can establish the truth of a general claim about the specific powers of a machine with infinite competence in Chomsky's sense, since any finite amount of evidence will have sample an infinitesimally small subset of possible behaviours for the machine. That implies that the evaluation of an explanatory theory for such a type of machine has to be much more indirect, and always in comparison with rivals that do better or worse over an extended period of research. This idea comes from Lakatos (1980). (I have omitted most of the details of the theories of Popper and Lakatos, for the sake of brevity.) A consequence of these arguments is that philosophers or scientists who propose that behavioural and other evidence can be the basis of 'Inference to the best explanation' of how human minds work are misguided. The most you can ever know is that one explanation is superior to others that have been suggested so far, given the facts so far found to be in need of explanation. Either new observations that need to be explained, or a new proposed explanatory theory can turn the best explanation into the second best, or worse.

NOTE: I argued in a paper published in Radical Philosophy 1976 "What are the aims of science" republished as chapter 2 of my 1978 book, that some of the deepest scientific advances have been ontology extensions postulating that something is **possible**, or that a class is capable of having instances. For that, the discovery of even a single case is conclusive proof, though subject to revision in case the example turns out to have been misdescribed -- a problem with all empirical evidence. As far as I am aware, neither Popper, nor his student Lakatos, ever considered such cases. A Nobel prize winning physicist, who read a draft of the paper agreed with me however: Anthony J Leggett, author of *The Problems of Physics* OUP 1987

In the case of computational systems, as in mathematics, a standard proof that something is possible is construction of a single example. Computer programmers are constantly demonstrating by example that forms of computation are possible that had never previously been considered, as illustrated by Turing's work in 1936.
End Digression]

Rice's Theorem

After I circulated a note about the black box experiments in my department, Martin Escardo pointed out that the argument above about what cannot be inferred from an examination of input and output tapes of a working TM demonstrates a special case of Rice's Theorem, which is often summarised by saying that no "interesting property" of a computational system can be proved by a machine observing its behaviour or examining its rules. There are many online presentations and discussions of Rice's theorem, e.g. http://en.wikipedia.org/wiki/Rice's_theorem

Rice's theorem (also known as the Rice-Myhill-Shapiro theorem) is stronger than my conclusion given above, since it states that even if the full specification of a Turing machine is given, no "interesting" property (using a technical definition of "interesting"!) can be proved from that specification, by another Turing machine, either by examining the rules or by running it on example inputs (which is just another way of trying to examine the rules). So my conclusion that you cannot discover the rules by observing the behaviours follows from Rice's theorem. But the proof given above, though long-winded, is less sophisticated than the proof of Rice's theorem, and may perhaps be more easily understood by intelligent non-mathematicians.

Marcin Milkowski has drawn my attention to the important paper by Zeigler (1974) on black-box testing, which I have not yet had time to study properly.

It should be emphasised that Rice's theorem and Zeigler's work make specific assumptions about the contents of the black boxes or Turing machines being investigated that I have argued elsewhere are not general enough to accommodate the variety of information processing architectures for biological organisms. For example, like many other authors they assume (if I have understood what I have read so far) that the systems being investigated have a fixed collection of possible discrete states and the internal processes are transitions between such states. However, those are inappropriate assumptions for biological physical machines and virtual machines, which may include continuous processes and many different interacting processes that are not synchronised, with new processes being spawned or old ones killed from time to time and new connections being set up between sub-systems when required. For more details see this discussion of Virtual Machine Functionalism (VMF): http://www.cs.bham.ac.uk/research/projects/cogaff/misc/vm-functionalism.html

(It is not clear to me whether the current state of theoretical computer science can accommodate all such machinery, although there have been steps in this direction, e.g. by Robin Milner and others. Certainly this specification goes beyond what can be modelled on any single Turing machine.)

Jump to Contents

Statistics Based Compression: Huge Google Engines (HGEs)
Expanded 25 Jun 2014

Since the days of discussion of giant lookup tables, a great deal has been learnt from the development of the world wide web and the technology for mining and using the information it contains. This shows that there are many alternatives to the use of giant lookup tables explicitly storing all the information. Various forms of compression, using patterns or rules inferred from the original data, can produce more compact systems that behave approximately as if they had stored all the information. The more compressed the new versions are the more they may diverge from the original data, for example in the probabilities attributed to various strings.

In the light of all those advances in technology it is clear that instead of the stark contrast between 'the real Turing Machine and a huge lookup table', or 'between a real human brain (or mind) and a huge lookup table', that used to exercise philosophers and AI theorists, we now have to consider a variety of increasingly sophisticated statistics-based machines that identify recurring patterns at various levels of abstraction and use them in mechanisms that avoid the memory requirements of a huge complete lookup table and can also speed up generation of plausible responses to inputs. So, instead of the contrast between identifying a human participant and being fooled by HLTs we also have to consider the possibility of interrogators being fooled by a collection of increasingly sophisticated intermediate HGEs (Huge Google Engines).

Instead of searching only for patterns that generate some larger but finite set of data by being instantiated by non-patterns (constants), we can allow patterns to be instantiated by patterns, thereby creating more complex patterns. For instance, allowing the variables in 'P and Q' to be instantiated by other patterns, e.g 'R and S', 'R or S' 'Not-R', we then have a finite (recursive) specification for an infinite set of patterns. In that case a learning machine encountering a lot of data could search in a space of increasingly complex patterns for an economical encoding of the data. Sometimes this allows a very large set of possible instances to be accommodated very economically (e.g. the infinite set of integers, or the infinite set of fractions, i.e. ratios of integers).

This can be the basis of a type of learning machine that finds re-usable patterns, constantly searching for the smallest pattern specification with infinite power that accommodates all its information records so far. A more general machine could allow statistical variation in some of its patterns to accommodate learning in a non-deterministic or partially understood environment, or for unreliable sensors, though it would probably need to use some sort of theory of the nature of the environment as a starting point. Such an innate theory could be a (lucky) product of natural selection over millions of generations.

As far as I am aware the most sophisticated theoretical exposition and working implementation of these ideas can be found in Jürgen Schmidhuber's work http://www.idsia.ch/~juergen/goedelmachine.html I don't yet have a deep understanding of this work, but as far as I know it cannot model the processes of discovery leading to Euclid's *Elements*, and cannot produce visual mechanisms with the sorts of powers discussed in http://www.cs.bham.ac.uk/research/projects/cogaff/misc/vision partly because it does not start with the required encoding of products of prior evolution.

All these design possibilities increase the variety of mechanisms available for fooling an interrogator attempting to tell what's inside a black box that responds to probing input signals. And instead of those deception mechanisms having to be carefully designed by highly intelligent programmers they can acquire much of their deception ability from all the gems and dross produced by humans in various formats that can be stored on the internet.

Of course, if and when these sorts of mechanisms begin to approach the sorts of intelligence found in humans and other animals, including elephants, squirrels, nest-building birds, hunting mammals, and cetaceans, including matching the developmental trajectories of humans and other altricial species we'll have a basis for saying that we know how to build a variety of types of intelligent machine on the basis of deep theories about the requirements and about mechanisms that are able to satisfy the requirements.

The evidence so far is that even with the additional sophistication provided by new more powerful hardware and statistical learning technology, the main result is so far only a marginal improvement in ability to behave like a human interlocutor. One worrying concern is that when we move beyond using

the technology for innocent games like the turing test and try to use data-mining techniques to provide real answers to real questions, instead of doing the deep scientific and engineering research required (but not guaranteed) to find answers, we may end up foolishly making serious use of devices that are little better than very deceptive chatbots, that don't know what they are telling us or why. Getting beyond that will require making use of a rich theory of architectural requirements for intelligent systems. Some starting points are suggested in CogAff papers and ideas for Virtual Machine Functionalism.

Because of spectacular successes in very constrained tests, the use of statistical learning in AI systems of many kinds has received an enormous amount of positive publicity. Many of the enthusiasts for such methods (surveyed in Clark (2013) have not asked enough questions about what such systems cannot learn, or do.

TO BE CONTINUED

Epilogue: Searle's Chinese Chatbot

Clearly this discussion has implications for John Searle's famous 'Chinese Room' thought experiment, which is based on the assumption that black box interrogation could give compelling evidence that understanding is going on even when it isn't. The discussions, as far as I recall, all failed to make the point that no amount of such testing could produce compelling evidence for anyone who understands the arguments presented above. Likewise many derivatives of Searle's thought experiment.
http://en.wikipedia.org/wiki/Chinese_room

This just reinforces the need to move away from behaviouristic science to focus on a search for deep explanatory theories, as in the Meta-Morphogenesis project.

THANKS

- To my colleague here in Birmingham, Martin Escardo, who responded to a query by pointing out that my argument for the impossibility of a behavioural test for a Universal Turing Machine was just a special case of Rice's Theorem (summarised and/or discussed on many web sites), e.g. http://en.wikipedia.org/wiki/Rice's_theorem
- To Marcin Milkowski for drawing my attention to the important paper by Ziegler (1974) which seems to be very relevant, but which I have not yet studied fully.
- To Harold Thimbleby for useful comments.

Jump to Table of Contents.

References (Copied from another paper: Pruning required)

- Mary-Ann Russon gave a useful summary of an interview on this topic by phone and email here: http://www.ibtimes.co.uk/why-turing-test-not-adequate-way-calculate-artificial-intelligence-1452120 Another journalist completely screwed up what I wrote to him, but later corrected his report and apologised.

- http://www.cs.bham.ac.uk/research/projects/cogaff/07.html#717
  Jackie Chappell and Aaron Sloman,
  Natural and artificial meta-configured altricial information-processing systems,
  International Journal of Unconventional Computing, 3, 3, 2007, pp. 211--239,

- http://www.cs.bham.ac.uk/research/cogaff/05.html#200502
  Aaron Sloman and Jackie Chappell, The Altricial-Precocial Spectrum for Robots,
  in *Proceedings IJCAI'05*, Edinburgh 2005, pp. 1187--1192,

- Andy Clark (2013)
  Whatever next? Predictive brains, situated agents, and the future of cognitive science, in
  Behavioral and Brain Sciences, 36, 3, pp. 1--24,

- J.J. Gibson, 1966, *The Senses Considered as Perceptual Systems*,
  Houghton Mifflin, Boston,

- J. J. Gibson, 1979, *The Ecological Approach to Visual Perception*,
  Houghton Mifflin, Boston, MA,

- Imre Lakatos, 1980
  Falsification and the methodology of scientific research programmes, in
  Philosophical papers, Vol I,
  Eds. J. Worrall and G. Currie, Cambridge University Press, pp. 8--101,

- John McCarthy, 2008 (on his web site since 1996)
  The Well-Designed Child,
  Artificial Intelligence, 172, 18, pp.2003--2014,
  http://www-formal.stanford.edu/jmc/child.html

- Karl R. Popper, *The logic of scientific discovery*, Routledge, London, 1934,

- http://www.cs.bham.ac.uk/research/projects/cosy/papers#tr0802
  Aaron Sloman, 2008,
  Kantian Philosophy of Mathematics and Young Robots, in
  Intelligent Computer Mathematics,
  Eds. Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M. and Wiedijk, F.,
  LLNCS no 5144, Springer, pp. 558--573,

- http://www.cs.bham.ac.uk/research/projects/cogaff/10.html#1001
  Aaron Sloman, 2010,
  If Learning Maths Requires a Teacher, Where did the First Teachers Come From?,
  Proc. Int. Symp. on Mathematical Practice and Cognition, AISB 2010 Convention,
  Eds. Alison Pease, Markus Guhe and Alan Smaill, pp. 30--39, ISBN 1902956931,

- Aaron Sloman, 2011,
  What's vision for, and how does it work?
  From Marr (and earlier) to Gibson and Beyond,
  http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#talk93,
  Online tutorial presentation, also at http://www.slideshare.net/asloman/

- http://www.cs.bham.ac.uk/research/projects/cogaff/03.html#200302
  Aaron Sloman and Ron L. Chrisley, 2003,
  Virtual machines and consciousness,
  Journal of Consciousness Studies, 10, 4-5, pp. 113--172,

- http://www.cs.bham.ac.uk/research/projects/cogaff/07.html#716
  Aaron Sloman, 2007,
  "Why Some Machines May Need Qualia and How They Can Have Them:
  Including a Demanding New Turing Test for Robot Philosophers,"
  AI and Consciousness: Theoretical Foundations and Current Approaches
  AAAI Fall Symposium 2007, Technical Report FS-07-01,
  Eds. A. Chella and R. Manzotti, AAAI Press, Menlo Park, CA, pp. 9--16,

- http://www.cs.bham.ac.uk/research/projects/cogaff/09.html#910
  Aaron Sloman, 2010,
  An Alternative to Working on Machine Consciousness,
  Int. J. Of Machine Consciousness, 2, 1, pp. 1--18
  (With commentaries and reply by author.)

- Aaron Sloman, (2012). Is education research a form of alchemy?, in
  Association for Learning Technology Newsletter, June, 2012, 27,
  http://www.cs.bham.ac.uk/research/projects/cogaff/misc/alchemy/

- http://www.cs.bham.ac.uk/research/projects/cogaff/11.html#1106c
  Aaron Sloman, The Mythical Turing Test, 2013, in
  Alan Turing - His Work and Impact,
  Eds. S. B. Cooper and J. van Leeuwen, Elsevier, Amsterdam, pp. 606--611,
  (This is a revised version of a subset of Sloman 2007.)

- A. M. Turing, (1950) Computing machinery and intelligence,
  Mind, 59, pp. 433--460, 1950,
  (reprinted in many collections, e.g. E.A. Feigenbaum and J. Feldman (eds)
  Computers and Thought McGraw-Hill, New York, 1963, 11--35),
  WARNING: some of the online and published copies of this paper have errors,
  including claiming that computers will have 109 rather than 10^9 bits
  of memory. Anyone who blindly copies that error cannot be trusted as a commentator.

- A. M. Turing, (1952), The Chemical Basis Of Morphogenesis, in
  Phil. Trans. R. Soc. London B 237, 237, pp. 37--72,
  (and reprinted in the 2013 Elsevier collection.)

- Bernard P. Zeigler, 1974,
  A Conceptual Basis For Modelling And Simulation, *Int J General Systems*,
  1, 4, pp. 213--228, http://dx.doi.org/10.1080/03081077408960781

-

-

-

-
-

Document History
Installed: 15 Jun 2014
Last updated: 22 Jun 2014; 25 Jun 2014; 5 Jul 2014; 26 Aug 2014

Maintained by
Aaron Sloman
School of Computer Science
The University of Birmingham