Proceedings of the Towards a Comprehensive Intelligence Test (TCIT): Reconsidering the Turing Test for the 21st Century Symposium,
Aladdin Ayesh, Mark Bishop, Luciano Floridi, and Kevin Warwick (Ed.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

The original test: it's harder than you might think

Darren Abramson

January 11, 2010

Abstract

In this paper I argue against the view that judges can easily distinguish between machines and humans in the Turing test. I show that a particular criticism of this form, as yet unrefuted in the literature on the test, is unsuccessful. In doing so, the paper presents a counterexample to the criticism of the test. I then identify open questions in the philosophy of computer science and demonstrate that the possibility of constructing the presented counterexample to the criticism of the test is reducible to these open questions.

1 Subcognition and the Turing test

This part of the paper involves a short demonstration, requiring the cooperation of able readers.

> Please bring your hands together, palsm pressed together, as if you were praying, touching the fingertips of your left hand with the corresponding fingertips of your of your right hand. Fold down your two middle fingers - and only your two middle fingers - so that the middle knuckles of both come together. (The tips of your thumbs, index, ring and pinky [little] fingers should be still be together.) Now, move your other fingers one at a time and report what happens. (from French (2000))

Can you move your ring fingers? Did you know you wouldn't be able to? Finally, what would happen if you asked this question in the Turing test of the two subjects being interviewed? Before we get to his argument, let us notice what French's demonstration does for our intuitions. Surely NO programmer could ever anticipate every such question, leaving us a toolbox for easily distinguishing human from machine.

In an earlier writing (French, 1990), Robert French imagines applying myriad results from cognitive psychology to the Turing test, by asking questions of the candidates concerning word similarity and semantic appropriateness of novel word strings. French concludes that the physical experience of human beings radically affects cognitive processing in ways that are revealed by these so-called 'subcognitive' questions. I have no argument with *this* claim. However, he goes further to make claims about the necessary relationship between the ability to answer these questions and having had certain experiences.

French suggests asking the two subjects in the Turing test even more salient questions, which will reveal not just cognitive differences which *depend* on physical differences, but will *directly* reveal physical differences (French, 2000) . For example, the subjects can be asked to observe novel properties of their own physiology, which they may not even be aware of until being asked, as we just saw. French anticipates objections that either these performative questions, or subcognitive questions, are not fair game in the Turing test scenario. I have no such objections.

French's claim, in short, is

> . . . as a test of general intelligence, the Turing test is not particularly appropriate precisely because it so hard: it tests not for intelligence, in general, but, rather, for culturally oriented human intelligence. In order to pass it, a machine would have to experience the world in essentially the same manner as we humans had, and, in order to do this, it would have to have a body and a set of experiences very similar to our own. And it is this that would make it virtually impossible for any machine to actually pass the Turing test. (French, 2000, 339).

Rather than rehearse the various subcognitive and embodiment-related questions which French mentions, I offer the following well constrained example which is both illustrative of his argument, and makes clear the issue in question.

> Consider, for example, the question: 'does freshly baked bread smell nicer than a freshly mowed lawn?' A machine that had never smelled either baking bread or a newly mowed lawn would have a great deal of trouble answering this question, unless, of course, it had specifically programmed to answer that particular question. But there are infinitely many such questions that humans can answer immediately because they can make a judgement, based on actual physical experience, about the degree of pleasure associated with each. Further, on average, most people within a particular culture will respond to this type of question in a similar manner. It is this fact that will be used to infallibly trip up the computer. (French, 2000, 334-335).

It is worth noticing that French's characterization of the only reasonable method for programming machines that pass the Turing test is explicitly rejected by Turing. I have shown elsewhere (citation suppressed) that Turing's presentation of the construction of 'learning machines' is motivated precisely by his rejection that computers can only do what we explicitly tell them to – witness the lengthy discussions of Lady Lovelace's objection, for instance. As we will see, the philosophy of computer science has already done a great deal to make issues about what implemented computers can do quite precise.

French elsewhere emphasizes the modal nature of his claim that in order to be prepared to answer certain questions, minds must physically interact in certain ways with the environment. He claims that being prepared to answer questions at the subcognitive level "[is] the product of a lifetime of interaction with the world which *necessarily involves* human sense organs, their location on the body, their sensitivity to various stimuli, etc" (French (1990, 62); emphasis in original).

French's argument is as follows. Necessarily, if you can appropriately answer subcognitive questions then you have had a lifetime of physical interaction with the world. Computers haven't had this lifetime of interaction. Therefore, by modus tollens, computers can't appropriately answer subcognitive questions, and are therefore unable to pass the Turing test. Now, if French's argument is sound, then Descartes has a new tool for allaying his hyperbolic doubt at the end of the first Meditation. Consider the first premise: necessarily, if you can, for example, describe preferences for some smells as opposed to others, then you actually experienced physical interaction with the environment. Forget about hyperbolic doubt – Descartes merely needs to ask himself a few questions about word association, favorite smells, and hand gestures, and he knows there is no evil demon deceiving him about the nature of his existence!

French's argument fails. There surely are possible worlds which contain only these two beings: Descartes and an evil demon. Descartes lives his life as a brain in a vat, and the evil demon makes Descartes think he has physical interaction with the world, when in fact he does not. This vat-Descartes could, presumably, pass a Turing test (or at least a culturally appropriate one) which included all sorts of subcognitive questions.

Someone sympathetic to the subcognitive objection to the Turing test will complain that I have only shown that a metaphysically possible brain can pass the Turing test – this is far from proving that a machine that had not experienced the world as we do could similarly pass it. I will now give a recipe for creating machines which have not experienced the world as we have, yet are able to answer subcognitive questions, and pass the Turing test.

2 How To Build a Machine That Passes the Turing test

1. Pick a person. Draw an imaginary line around their brain and nervous system.

2. Code up a complete model of everything inside the line. That is, for completely deterministic portions of the tissue involved, construct a model which behaves exactly as the tissue does. For non-deterministic portions, have the model behave in identically probabilistic fashion.

3. Code up a model of the effect the following stimuli have on the nervous system just modelled:

   (a) being invited to participate in a Turing test;

   (b) arriving at the test site; (If the model outputs some other decision than participating, erase the session, go back to the first model, and keep simulating until a decision is made to participate. If the model turns out to be recalcitrant, pick someone more cooperative to model.)

   (c) sitting down at the terminal, and seeing questions flashed on a screen in a room just like the one in which the real human subject is sitting.

4. Apply the above stimuli to the model. Copy the model to the computer in the Turing test scenario.

5. For each question, deliver inputs to the brain model in the form of simulated inputs to the nervous system; track simulated efferent outputs to motor neurons, thereby tracking simulated key presses, and send the output to the questioner.

Notice that the computer program described will not only pass the Turing test, answering any variety of subcognitive questions appropriately, but in the case that the program is competing against the 'source' for the model, it *will be indistinguishable from the competing real human being.* The reason for this should be obvious. We have imagined coding a program which faithfully reproduces the outputs of a human nervous system. Then, we have imagined stimulating this simulated nervous system with the inputs a nervous system would receive in taking the Turing test. The only source of difference we might find would be in errors in either simulating the input, or the nervous system itself, and we are supposing no such errors have been made.

Think about the task I asked you to do a few minutes ago. If we can and do write the program I have described, it will respond in just the same we do – with a message something like 'wow, I can't move my ring fingers. I didn't expect that'. This doesn't threaten our intuitions, though. By simulating the muscle contractions, bone movements, and cartilage tension of arms, and sending appropriate inputs to our simulated brain, we received just the same response we would from a real brain connected to real muscles, bones, and cartilage.

Of course, it is an open question whether or not a machine can ever be built, as described, that passes the Turing test. In the remainder of this paper, I will show how we might formulate this question more clearly, and the way in which central issues in the philosophy of computer science bear on it.

3 Really Hard Questions For Machines

It should be emphasized that the program described above can not only pass the Turing test when asked subcognitive questions, it can even – if the programming is done well – pass questions that are so hard, French excuses them from fair game during the Turing test. In his 1990 article, French supposes that the judge prepares him or herself by doing a series of experiments on the psychological phenomenon of 'associative priming' (explained below).

The judge has available a series of macros for sending strings at a preset pace over the terminal hookups in the test, and a piece of software scanning responses from subjects on the screen and measuring latency. Having practiced earlier with a host of human subjects in similar terminal hookups, the judge can come in with statistically reliable predictions on variable latencies for responding to key presses for, say, identifying words from non-words.

It turns out that people respond faster in identifying a string of letters as a word if it is preceded by a semantically related word. As French tells us, "[if] . . . the item 'butter' is preceded by the word 'bread', it would take significantly less time to recognize that 'butter' was a word than had an unassociate word like 'dog' or a nonsense word preceded it" (French (1990), 57). The familiar conclusion, that no computer could pass the Turing test, is presumed to follow from the ability of the judge to ask such questions.

> The machine would invariably fail this type of test because there is no a priori way of determining associative strengths (i.e., a measure of how easy it is for once

> concept to activate another) between *all* possible concepts. Virtually the only way a machine could determine, even on average, all of the associative strengths is to have experienced the world as the human candidate and the interviewees had. (French (1990), 58)

Precisely what is left out by the word 'virtually' here? Perhaps French means that, barring exceptional luck and accident, there is no way to build a machine that one expects to pass such tests. He supposes that a critic might object to these questions, since they don't really seem in the spirit of the test. After all, these are explicitly subcognitive questions, and the Turing test is intended to make cognitive distinctions. So, French "obligingly disallows" (ibid.) such questions. I will argue now that no such obligation is necessary. Machines designed as I have described can, if certain programming choices are made, pass these psychometric tests also. Also, by seeing that this is true, we can help avoid a mistake that comes up from time to time in discussions of cognition and computation.

Cummins et al. discuss ways in which you might decide between connectionist and language of thought implementations of cognitive architecture in people. They distinguish between 'primary' and 'incidental' effects of algorithms (Cummins et al., 2001) . Primary effects refer to the computational task a machine or organism accomplishes, while incidental effects are all those facts which may be true or false without affecting whether the task is performed. If one machine emits a beep while factoring natural numbers, but the other emits a squawk, then they might have the same primary effects (factoring numbers correctly) but different incidental effects.

There might be connectionist and rule-based parsers that recognize precisely the same strings, but they argue that there is reason to think that connectionist parsers would be revealed by little or absent growth in processing time compared to input size. However, rule-based parsers, they say, would have some positive correlation between input and processing time.

Here is their summary of the possible sources of difference between two systems in their incidental effects.

> One important source of incidental effects thus lies in the algorithm producing the primary effects. But systems that implement the *same* algorithm may exhibit different incidental effects. If the underlying hardware is sufficiently different, they may operate at greatly different speeds, as is familiar to anyone who has run the same program on an Intel 486/50 MHz [computer] and on a Pentium III [computer] running at 750 MHz. A significant difference in response times between two systems, in other words, does not entail that they implement different algorithms; it may be a direct result of a difference at the implementation level. Hence, two systems that have identical primary effects may exhibit different incidental effects either because they compute different algorithms or because the identical algorithms they compute are implemented in different hardware. (Cummins et al. (2001), 174).

Notice that these authors are not saying that if architecture and/or algorithm are different then there will necessarily be a difference in incidental effects. In fact, they go on to say that one cannot in general infer from sameness of incidental effects to sameness of functional architecture, which presumable includes both algorithm and implementation.

However, they also appeal to our idea that if you have two completely different functional systems, surely timing will generally go askew between them. Let's look briefly at a counterexample.

I will stipulate the following definition of emulation: one computer emulates another just in case they display both the same primary effects *and* all the same secondary effects, on the definitions offered by above. Furthermore, I will limit secondary effects to those that can be detected by a judge in the context of the Turing test. So, latency between stimuli and responses will count as secondary effects, but the warmth of the processor of the computer being tested will not.

There was a time when video games from one PC platform were unplayable on another. However, in dealing with rapidly increasing processor speeds, programmers tied event timings to some absolute measure instead of clock ticks. With a computer equivalent to a universal Turing machine, the ability to simulate exactly the *timing* of another computational system depends only on the speed at which it operates and the number of instructions it can execute per cycle. Routinely, popular web browsers can display interactive windows in which entire computer architectures in production less than a decade ago are emulated in the sense defined here.[1]

In the next section I will consider objections to the claim that any computer can be built, for the purpose of passing the Turing test, in the manner I have described.

4 Objections

4.1 Brains and their environment cannot be separated, even in principle.

Recent opinions in the foundations of cognitive science suggest such a statement.[2] Consider the idea that human minds are intrinsically embodied and embedded. A strength of French's argument, one might claim, is that it takes advantage of this property of persons. If this objection is taken to mean something like 'if you isolate a brain from body and environment then you no longer have a person', I am likely to agree. However, we are not trying to reproduce people here – we are merely trying to simulate part of a body.

If the objection is taken to mean something stronger, like 'one cannot in principle isolate and emulate a biological subsystem, namely the brain and spinal cord' then objection appears weak. No doubt isolating and emulating any biological subsystem is prohibitively difficult with present technology. For all I know it is impossible – I will talk in great detail below about particular reasons doing this might be impossible below. However, despite continuous interaction between neurons and the rest of the world, it seems at least plausible that future scientists could arbitrarily pick a physical boundary – say synaptic junctions at distal nerve endings – and model everything inside.

4.2 We cannot, in principle, model input to the nervous system.

There are a few problems involved once we have our nervous system modelled. We must be able to specify a physical scenario in somewhat large scale terms, and then translate that scenario into simulated effects on individual synapses. Again, this is a difficult task. A few examples of imaginary beings capable of doing this come to mind: Descartes's evil demon, and the managers of the fictional 'Matrix' of the popular movie by that name. Let us consider this objection in isolation from the one above. Suppose that we are only worried about emulating inputs to a nervous system, whether the nervous system itself is real or emulated. Someone who objects to the possibility of accurately emulating the input that particular environments provide to the nervous system is committed to the success of a judge in what I will call the *Wachowski test*.

The Wachowski test is a lot like the Turing test. However, instead of trying to figure out which chat window is connected to a machine and which is connected to a person, the goal of the judge is to distinguish a person experiencing an *emulated* terminal, entirely computer generated and delivered directly to the nervous system, from the person who really is sitting at terminal typing responses. Again, for all I know, no one, in tandem with the best engineering possible, could produce a judge-fooling entry for the Wachowski Test. Two observations should suffice for rejecting this objection against my counterexample to French's argument, however. First, it is at least possible that no judge can beat the Wachowski test for sufficiently advanced emulations. Second, and now I am considering both objections presented so far, French never mentions *any* reason to think we can't simulate either the environment or the nervous system.

4.3 The Identity Objection

I find this objection the most interesting. It avoids appealing to impossibilities which need to be established empirically, and are far beyond any reasoning we can make from present physics and neuroscience.

Suppose we have chosen to simulate a particular person – say Robert French himself. We conduct the Turing test between Prof. French at one terminal, and the simulation plugged into the environment emulator program at the other end. At the second terminal, text inputs and outputs are received and produced by a program which translates to and from emulated stimuli and responses of the emulated brain.

Recall French's conclusion: to be able to pass the Turing test requires a lifetime of experience involving the world. An objector might claim that the computer in this case, by running an emulation of Robert French's brain *does* have a lifetime of experience involving the world.

While appealing, I argue that this objection – identifying the machine's experience with the experience of its modeling source – is illicit. This can be seen in a few ways.

First, if French's argument is intended to have a general conclusion, then it must deal with computers randomly selected from the space of all possible computers. It is true that given the space of possible computers complex enough to emulate a human brain, very few will produce Turing test-passing output. In fact, very few will produce recognizable outputs at all! However, there will be some which emulate my brain, others that emulate your brain, and so on for every other brain (that is, supposing other considerations discussed below are decided in a particular way). This is true, even granting that which computers simulate which brains will depend on the program we use for supplying simulated environments. So, while French's brain may play an epistemic role in programming a particular machine, the program itself hasn't had any experience – it waits quietly in Plato's heaven until we retrieve it using a neural scanner, or some other such putative device.[3]

Second, there is a strong sense in which the computer hasn't had any of the experiences which French has had. In a naive sense, the computer – if it had experiences at all – merely had those of being manafactured, plugged in, and then programmed. Its program doesn't even contain mention of interactions with the environment; only numerical solutions to differential equations, perhaps, modelling spike trains given certain stimuli.

Third, experiences presume an experiencer. If we are able to program a computer as I have described, does that imply that computational functionalism, as a theory of qualitative experience, is true? It doesn't seem so. Notice that if we are committed to the computer's sharing French's experiences, then we will be forced to discount the counterexample I have offered at the expense of presuming the truth of computational functionalism. Surely there are possible worlds where computers can emulate brains, and minds – which are the things which do the experiencing – exist as epiphenomenal immaterial souls.[4]

5 Reasons to Think That Brains (and/or environments) Can't Be Emulated

There are two broad sets of reasons which would render my counterexample to French's arguments a failure. First, there could be contingent limits on the computers we can build which would prevent us from every accurately modelling a brain. For example, we might not be smart enough to ever decode the complicated functions being computed by neurons. Or, a cataclysmic solar flare might incinerate the world before we finish working on our best models. These considerations are beside the point, though. French's argument is about the in-principle limits available for constructing machines that pass the Turing test.

More interesting are the possible failures of two related theses. The Church-Turing thesis says that all of the effectively computable functions are Turing computable. This means that all of the functions which a human being can compute by using unbounded time, paper, and pencil, and simple rules about writing, recognizing, and changing symbols, can be computed by a regular Turing-equivalent computer. We can consider an expansion of this thesis, and call it the Physical Church-Turing thesis (as, for example, Deutsch (1985) does). This says that all physically computable functions are Turing computable. Now, suppose that the thesis fails, and that human nervous systems compute functions which no Turing-equivalent computer can. Then no matter how much time, ingenuity, and material support are available, no counterexamples of the form I have presented are possible.

The second thesis is an extension of the Church-Turing thesis out of the realm of computability and into computational complexity. This thesis says that any function computable by a physical device can be computed by a Turing-equivalent computer with at most polynomial slowdown. This 'Extended Church-Turing thesis', as Shor (1988) calls it, might be false. If human nervous systems are counterexamples to this thesis, then for sufficiently long instances of the Turing test, our best simulations will eventually crawl to a halt.

If these modified Church-Turing theses are true, and the contingencies don't prevent us from modelling brains and nervous systems and relevant environments, then it seems that the counterexample I have offered to French's claim has real world possibility. French's claim, when understood in its presented strong modal sense is clearly false. We can modify his claim to become merely one about this world, namely that no machine we can actually build will pass the Turing test by failing subcognitive questioning. To justify this will require more than listing off questions which appeal to embodied experience. It will require either offering reasons to think that one of the variants of the Church-Turing thesis is false for relevant reasons. Or, it will involve discussing relevant contingencies which prevent us from accomplishing an impractical, nomological possibility.

I do not have an answer here as to whether the physical or extended Church-Turing thesis is true. I have argued previously that there are not, as yet, any convincing refutations of the possibility of building a machine that outcomputes a standard Turing machine (citation suppressed). As for the extended Church-Turing thesis, there has been considerably less interest in the philosophy of computer science in investigating whether physical objects might display complexity properties that standard computers don't, prompting one researcher to comment that there is "...an overemphasis on the concept of computability"' (Volchan, 2002, 61).

The in-principle possibility of building a machine that passes the Turing test in the manner described would follow from the conjunction of the physical and extended Church-Turing theses. Notice, though, that the failure of either for biological systems, or other parts of the Turing test arrangement, could render the putative machine I have described impossible. First, notice that any finite function is Turing computable, and therefore physically computable. Second, notice that an emulation of any finitely specified performance can be achieved: load all responses into a finite amount of random access memory, and have a clock provide desired delays. A central characteristic of the Turing test is its unbounded nature. As many have noticed, no preset, finite bound on the length of the test can be specified before conducting the test; doing so renders the test trivial (an early expression of this can be found in Sampson (1973)).

The philosophical discussion of the Church-Turing thesis, and related forms, is vast. Historical questions, such as whether Alan Turing or Alonzo Church intended their theses to apply to physical objects, for example, is disputed (a useful summary can be found in Copeland (2002)). Descriptions of machines that compute functions no Turing machine can compute, the existence of which are consistent with known physics, have been offered (many of these are summarized and critically evaluated in Shagrir and Pitowsky (2003); Shagrir (2004)). There may be no answer to the truth or falsity of these theses forthcoming. However, I hope to have shown that no argument that ignores these theses can convince us of whether or not a machine can be built that passes the Turing test and that, according to both Turing and Descartes, deserves to be considered a thinking thing.

Notes

[1]Due to copyright restrictions governing the systems emulated, websites offering such services come and go quickly. For a selection of such sites, one can Google the terms 'web based emulator.'

[2]For example, consider work supporting the so-called 'enactive theory of perception' as presented, for example, in (Noë, 2004).

[3]This reply relies on my own views on answers to presently open questions in the philosophy of computer science, identified by Turner and Eden (2009).

[4]I have avoided in this paper discussion of computational functionalism: the theory that having a mind is nothing more than implementing a particular computer. If true, then philosophy of mind can be *identified* with the philosophy of computer science.

References

Copeland, B. J. (Fall 2002). The Church-Turing Thesis. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*.

Cottingham, J. (1992). Cartesian dualism: theology, metaphysics, and science. In Cottingham, J., editor, *The Cambridge Companion to Descartes*. Cambridge University Press.

Cummins, R., Blackmon, J., Byrd, D., Poirier, P., Roth, M., and Schwarz, G. (2001). Systematicity and the cognition of structured domains. *Journal of Philosophy*, 98:167–185.

Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A*, 400:97–117.

French, R. (1990). Subcognition and the limits of the Turing Test. *Mind*, 99(393):53–65.

French, R. M. (2000). Peeking behind the screen: the unsuspected power of the standard Turing Test. *J. Exp. Theor. Artif. Intell.*, 12(3):331–340.

Noë, A. (2004). *Action in Perception*. MIT Press.

Sampson, G. (1973). In defence of Turing. *Mind*, 82(328):592–594.

Shagrir, O. (2004). Super-tasks, accelerating turing machines and uncomputability. *Theoretical Computer Science*, (317).

Shagrir, O. and Pitowsky, I. (2003). Physical hypercomputation and the church-turing thesis. *Minds and Machines*, 13:87–101.

Shor, P. W. (1988). Quantum computing. *Documenta Mathematica - Extra Volume - Proceedings of the International Congress of Mathematicians*, I:467–486.

Turner, R. and Eden, A. (2009). The philosophy of computer science. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Summer 2009 edition.

Volchan, S. B. (2002). What is a random sequence? *Mathematical Association of America Monthly*, 109:46–63.

Proceedings of the Towards a Comprehensive Intelligence Test (TCIT): Reconsidering the Turing Test for the 21st Century Symposium,
Aladdin Ayesh, Mark Bishop, Luciano Floridi, and Kevin Warwick (Ed.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Demo: Have a Chat with Sensitive Artificial Listeners

Marc Schröder, Sathish Pammi[1], Roddy Cowie, Gary McKeown[2], Hatice Gunes, Maja Pantic, Michel Valstar[3], Dirk Heylen, Mark ter Maat[4], Florian Eyben, Björn Schuller, Martin Wöllmer[5], Elisabetta Bevacqua, Catherine Pelachaud, Etienne de Sevin[6]

Abstract. Sensitive Artificial Listeners (SAL) are virtual dialogue partners based on audiovisual analysis and synthesis. Despite their very limited verbal understanding, they intend to engage the user in a conversation by paying attention to the user's emotions and non-verbal expressions. The SAL characters have their own emotionally defined personality, and attempt to drag the user towards their dominant emotion through a combination of verbal and non-verbal expression. The demonstrator shows the publicly available, fully autonomous SAL system.

1 INTRODUCTION

The SAL demo supports sustained emotionally-coloured machine-human interaction using non-verbal expression. The system aims to engage the user in a dialogue by paying attention to the user's non-verbal expression, and reacting accordingly. It focuses on the "soft skills" that humans naturally use to keep a conversation alive.

To simplify the challenge somewhat, the SAL system avoids task-oriented dialogue. Instead, it models the type of interaction found at parties: you listen to someone you want to chat with, and without really understanding much of what they are saying, you exhibit all the signs that are needed for them to continue talking to you. Similarly to the Rapport agent [2], SAL characters show non-verbal listener signals; in addition, they can also speak to engage the user in a simple dialogue. The approach has been test-run using Wizard of Oz setups at various stages of maturity [1]. This has allowed us to fine-tune the scripts used by the various characters, in order to react to the emotional state of the user in plausible ways despite the lack of verbal knowledge. The system is freely available from the project website `www.semaine-project.eu`.

2 THE DEMONSTRATION SETUP

In the SEMAINE demonstrator, one human user is sitting in front of a computer screen showing the face of an Embodied Conversational Agent (ECA). The user is wearing a headset for voice analysis and is recorded by a video camera for facial expression analysis. The ECA is speaking through loudspeakers, and is showing both verbal and non-verbal behaviour. A second screen shows a system monitor, displaying graphically the current information flow in the system.

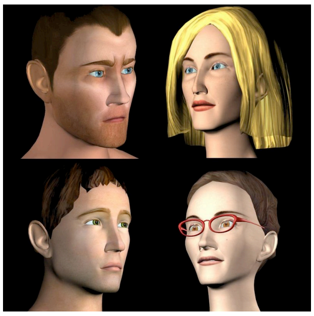

Figure 1. The four SAL characters represent the four quadrants of arousal-valence space: Spike is aggressive; Poppy is cheerful; Obadiah is gloomy; and Prudence is pragmatic.

The user can speak to one of the four SAL characters at a time (see Figure 1). Each character will try to sustain the conversation by being an active speaker and listener using multimodal verbal utterances and feedback signals. After a while, the user can switch to a different character. In the lab, sessions typically last for around 20 minutes; in the demo, much shorter sessions with changing users are anticipated.

Technically, the demonstrator system is a multimodal interactive system with components integrated across programming languages and operating systems by means of a standards-based framework for building emotion-oriented systems, the SEMAINE API [3]. Details on the technological setup and the individual processing components are described in a set of project deliverable reports available from the project website.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 211486 (SEMAINE).

REFERENCES

[1] E. Douglas-Cowie, R. Cowie, C. Cox, N. Amir, and D. Heylen, 'The sensitive artificial listener: an induction technique for generating emotionally coloured conversation', in *LREC2008 Workshop on Corpora for Research on Emotion and Affect*, pp. 1–4, Marrakech, Morocco, (2008).

[2] J. Gratch, N. Wang, J. Gerten, E. Fast, and R. Duffy, 'Creating rapport with virtual agents', in *Intelligent Virtual Agents*, 125–138, (2007). doi:10.1007/978-3-540-74997-4_12.

[3] M. Schröder, 'The SEMAINE API: towards a standards-based framework for building emotion-oriented systems', *Advances in Human-Computer Interaction*, **2009**(319406), (2009). doi:10.1155/2009/319406.

[1] DFKI GmbH, Saarbrücken, Germany, email: firstname.lastname@dfki.de
[2] Queen's University Belfast, UK, email: (r.cowie | g.mckeown)@qub.ac.uk
[3] Imperial College London, UK, email: firstname.lastname@imperial.ac.uk
[4] Universiteit Twente, The Netherlands, email: (d.k.j.heylen |maatm)@ewi.utwente.nl
[5] Technische Universität München, Germany, email: lastname@tum.de
[6] CNRS, Telecom Paristech, France, email: firstname.lastname@telecom-paristech.fr

Proceedings of the Towards a Comprehensive Intelligence Test (TCIT): Reconsidering the Turing Test for the 21st Century Symposium,
Aladdin Ayesh, Mark Bishop, Luciano Floridi, and Kevin Warwick (Ed.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

A proposed replacement for the Turing Test

Douglas A. Samuelson, InfoLogix, Inc., Annandale, VA USA

Abstract. As the Turing Test of intelligence relies on deception, we seek a more general test of intelligence that includes the Turing Test as a special case. We posit that essential characteristics include meta-learning (learning about learning), meta-generation (generation of rule generators), meta-assessment (assessment of assessment methods), and awareness of other beings' emotional states and reactions as well as their logical processes. Accordingly, in this brief position paper, we offer a new proposed test: An intelligent entity can learn how its actions increase or decrease other sentient beings' satisfaction or dissatisfaction with their current state and circumstances, and can then manipulate that information to its advantage

1 INTRODUCTION

The Turing Test of machine intelligence is brilliantly simple. When proposed, it represented a major improvement over more complicated theories that required complex philosophical assumptions. It is both logically solid and intuitively appealing: if the machine can somehow learn what abilities to display and not to display to convince a human experimenter that it strongly resembles a human, then it is, indeed, an entity most people would consider intelligent.

There are two problems with this definition. First, it is based on deception. This makes some of us uncomfortable, as we would like to consider as well a machine that can please us honestly. This is a special case of the second problem: the Turing definition is perhaps sufficient but clearly not necessary. Without making the assessment hopelessly complicated, we would like to broaden the test to include other ways to demonstrate intelligence.

2 PROBLEM STATEMENT

We seek a test for machine intelligence that is both practical and exhaustive: it should be both necessary and sufficient as a criterion for machine intelligence. The Turing Test fails the "necessary" criterion, as we can conceive of clearly intelligent non-human entities that would not pass the Turing Test.

In this author's view, the new test should:
-- incorporate learning, generation of rules, generation and assessment of generators;
-- incorporate awareness of others' emotions, where present, not just logic;
-- not be species-specific;
-- take into account that an intelligent entity may not be biological or electromechanical: organizations count too (for example, Hofstadter's "Aunt Sarah," the ant colony from *Goedel, Escher, Bach*);
-- avoid the "Searle's Chinese Room" paradox in which entities can appear to behave intelligently by following a script but lack other essential features, such as ability to adapt;
-- not require frequent human intervention and assistance to achieve and maintain high performance;
-- imply the Turing Test as a special case; and
-- remain both practical to apply and theoretically sound.

3 DISCUSSION

Recent research indicates that learning and adaptation are essential elements of our definition. Moreover, learning about learning, assessing means of assessment, and adapting ways of adapting – in short, recursive layered behaviors – now appear to be critical components of intelligence.

One counter-argument to the Chinese Room paradox is that it depends on an artificial constraint: it relies on the assumption that the experimenter not only cannot distinguish between the translations produced, but also cannot ask the translators to do so. In real life, asking experts to critique each other's expertise is common practice, and often most informative. Similarly, in the Turing Test situation, the experimenter could ask each entity to comment about whether the other seems human. The entities would be challenged not only to learn how to present their abilities to a human judge, but also to learn why they should try. That is, their rewards for trying to pass the test would not only influence their behavior in the test but also teach them that the experimenter wants to be deceived; we might then wish to teach them when deception is not desired. Thus we would place greater value on the machine that can not only learn to deceive us but also learn when not to do so.

This implies, in turn, an ability to learn not just how to solve the problem presented, nor even just how *not* to solve sub-problems in a way that gives away information better concealed, but also to learn about the experimenter's likes and dislikes. Just as with other people, we are likely to regard most highly not the fastest logical processor, but the entity that can most quickly learn to assess other sentient beings' emotional reactions to its actions and adjust its behavior accordingly, to its advantage.

We therefore come to the following proposed test.

4 PROPOSAL

To be considered intelligent, the entity demonstrates that it can learn how its actions increase or decrease other sentient beings' satisfaction or dissatisfaction with their current state and circumstances, and can then manipulate that information to its advantage.