Evaluating a human

Artificial intelligence is no longer the preserve of academia and science fiction, explains **Julian Bucknall**

Back in the late '40s, Alan Turing, fresh from his success as part of the team that cracked the Enigma machine, turned his thoughts to machine intelligence. In particular, he considered the question: 'Can machines think?' In 1950, he published a paper called Computing Machinery and Intelligence in the journal Mind that summarised his conclusions. This paper became famous for a simple test he'd devised: the Turing Test.

The problem he found in particular was the word 'think' in his original question. What is thinking and how do we recognise it? Can we construct an empirical test that conclusively proves that thinking is going on? He was of the opinion that the term was too ambiguous and came at the problem from another angle by considering a party game called the Imitation Game.

Parlour games

In the Imitation Game, a man and a woman go into separate rooms. The guests don't know who is in each room. The guests send the subjects written questions and receive answers back (in Turing's day these were typewritten so that no clues could be found by trying to analyse the handwriting; today we can imagine email or Twitter as the medium). From the series of questions and answers from each subject, the guests try to work out who is the man and which the woman. The subjects try to muddy the waters by pretending to be each other. The first part of Figure 1 shows this game. Turing wondered what would happen if we replaced the man or the woman with some machine intelligence. Would the guests guess the machine versus human more accurately than they would the man versus the woman? Turing's insight was to rephrase 'Can machines think?' into a more general 'Can machines imitate how humans think?' This is the original Turing Test, and is shown in the second part of Figure 1.

Over the years, this Turing Test has changed into the simpler test we know today: can we determine whether the entity at the end of a communications link is computer program or human, just by asking questions and receiving answers? It's is still a variant of the Imitation Game, but now it's much simpler – possibly too simple. ▸

The first program to try to pass the Turing Test was a program called ELIZA, a program that pretended to be an empathic or non-directional psychotherapist, written in the period 1964-1966. ELIZA essentially parsed natural language into keywords and then, through pattern matching, used those keywords as input to a script. The script (and the database of keywords and the subject domain) was fairly small, but nevertheless ELIZA managed to fool some people who used it. The reason for using a psychotherapy as the domain is that it lends itself to being able to respond to statements with questions that repeat the statement ("I like the colour blue." "Why do you like the colour blue?") without immediately alerting the human that the therapist on the other end has no applicable real-world knowledge or understanding.

ELIZA was so successful that its techniques were used to improve the interaction in various computer games, especially early ones where the interface was through typed commands.

Chatterbots

Despite its simple nature, ELIZA formed the basis of a grand procession of programs designed and written to try to pass the Turing Test. The next such was known as PARRY (written in 1972, and designed to be a paranoid schizophrenic), and they spawned a whole series of more and more sophisticated conversational programs called chatterbots. These programs use the same techniques as ELIZA to parse language and to identify keywords that can then be used to further the conversation, either in conjunction with an internal database of facts and information, or with the ability to use those keywords in scripted responses. These responses give the illusion to the human counterpart that the conversation is moving forward meaningfully.

Consequently, provided the subject domain is restricted enough, chatterbots can and do serve as initial points of contact for support. For example, PayPal currently has an automated online 'customer service rep' it calls Sarah. Using the chat program is quite uncanny. As you can see in Figure 2, the answer to my question ("How do I see the payments I made in January") appeared instantly and the natural language evaluation processing is extremely efficient. Notice that the word 'payment' is recognised and the more accurate 'transaction' is used in the reply). Such automated online assistants are available 24/7, and are helping reduce the loads on normal call centres. Aetna, a health insurer in the US, estimates that 'Ann', the automated assistant for its website, has reduced calls to the tech support help desk by 29 per cent.

Of course, there are downsides to chatterbots as well. It's fairly easy to write chatterbots to spam or advertise in chat rooms while pretending to be human participants. Worse still are those that attempt to cajole their human counterparts into revealing personal information, like account numbers.

Depth of knowledge

In reality though, chatterbots are still too simple for us to be fooled for long. They don't have the depth of day-to-day knowledge that makes conversations interesting in the real world, and any conversation over a few minutes reveals that paucity of information.

A more recent development was the episode of the quiz show Jeopardy in which two human contestants played against IBM's Watson computer (although given the fact that Watson has 2,880 processor cores and 16 terabytes of RAM, perhaps 'computer' is too simple a term). Watson was not only programmed with a natural language interface that could parse and understand the quiz questions, but also had a four-terabyte

Spotlight on… Playing Civilization

Writing programs to play games has had a long history. The most famous game to be programmed is, of course, chess, with the IBM computer Deep Blue finally beating a Grand Master (Garry Kasparov) in 1997.

One recent addition to the list of game-playing programs is one that plays turn-based strategy title Civilization. In a 2011 paper, Learning to Win by Reading Manuals in a Monte-Carlo Framework, Branavan et al investigated improving a program that played Civilization II by allowing it to parse and use the manual, which details various strategies players can apply to the game.

The original program used Monte-Carlo techniques (essentially generating a stream of random numbers and using those to affect in-game decisions) to simulate various game scenarios and then to select the optimal ones. The researchers then wondered if there was a way to direct the search for a best scenario by using strategies detailed in the manual. A neural network was designed that took the current state of play, a proposed move and the parsed manual, and then trained itself. Over the course of a game, the network would learn when to apply the various strategies, and overall, the game-playing program would improve its score from the non-manual-reading program by 27 per cent, winning over 78 per cent of its games.

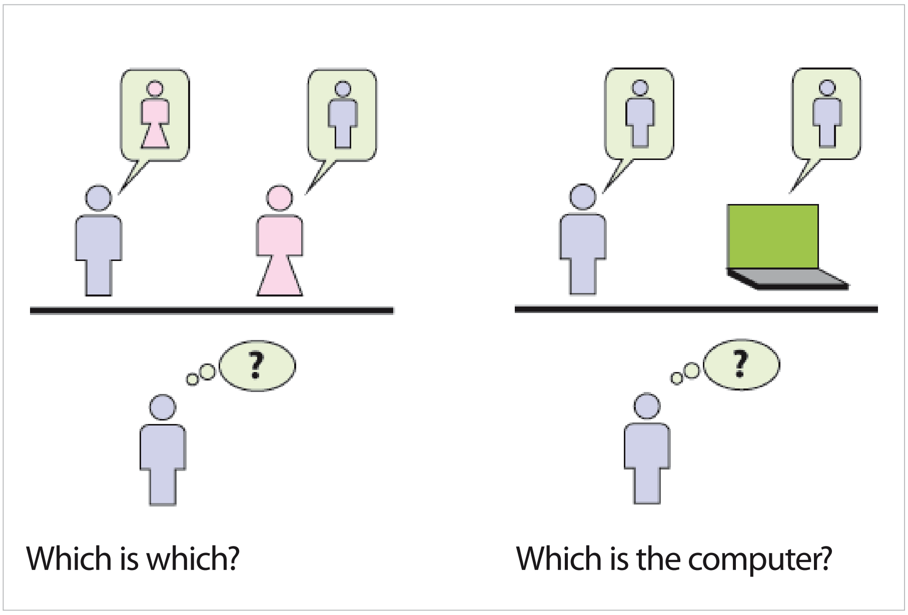
The strange thing is, feeding in text from pages of the Wall Street Journal also improved the program's scores. Not to the same extent as when the real manual was used, but definite improvements were seen. ■

Representing knowledge

One of the biggest problems that AI researchers must solve is how to represent information about the world, or at least about the specific domain for the AI under question. An AI has to represent objects, their properties, their behaviours, any categories they may belong to, and the relations between one object and another.

Part of the problem is the sheer breadth of knowledge that needs to be represented (IBM's Watson used four terabytes of data). The average person has learned billions of individual facts by the time they've reached adulthood and programming that 'by hand' is inconceivable. This is why most ontological databases must be built through the AI learning from parsing resources like encyclopedias and news sites.

Another part of the problem is that some of people's knowledge is intuitive, and not necessarily factual. Why is the Mona Lisa such an amazing work of art , what makes Beethoven's Fifth a great symphony, and so on. ■



▲ **Figure 1: An illustration of the Imitation Game and the Turing Test.**

database of structured and unstructured information from encyclopedias (including Wikipedia), dictionaries, thesauruses, and other ontologies. In essence, Watson has 'knowledge' about many things and the software to query that knowledge, to form hypotheses about that knowledge, and to apply those hypotheses to the questions posed.

Parsing natural language

Although Watson may be seen as the current best contender for passing the Turing Test, there are still issues with the natural language interface – perhaps the hardest part of the software to write. One of the questions asked (or rather, given the nature of the quiz show, one of the answers for which the contestants had to provide the question) was: "Its largest airport was named for a World War II hero; its second largest, for a World War II battle", and Watson failed to parse the sentence properly, especially the part after the semicolon, causing it to reply with "What is Toronto?" when the answer should have been "What is Chicago?".

Natural language can be parsed in two ways: either through a strict semantic analysis and continually improving that analysis, or through a looser analysis and then using statistical methods to improve the result. This kind of algorithm is used by Google Translate: by using a huge corpus of texts and translations by human linguists, the quality of on-demand translations can be improved greatly. Google Translate uses the power of the crowd to translate text rather than strict language algorithms, but it's a useful application of AI.

One of the problems with the Turing Test is that it invites programs with ever more complex conversation routines that are just designed to fool a human counterpart. Although researchers are investigating how to create a 'strong AI' that we can converse with, more research is being done on 'specific AI' or 'domain-bound AI' – artificial intelligence limited to a specific field of study. Real advances are being made here, to the extent that we no longer think of these solutions as AI. Indeed, there's more of a move to view AI research as research into problems whose solutions we don't yet know how to write.

An example of such specificity in AI research is face detection in images. Yes, it's been solved now, but it was only 2001 when Paul Viola and Michael Jones published their paper on how to approach the problem. A decade later, we have point-and-shoot cameras that can find a face in the field of view, then focus and expose for it. Fifteen years ago or earlier, the face detection problem would have been considered AI, and now we have cameras that can do the work in real time. AI is a concept that shifts its own goalposts.

Neural networks

Many specific-AI systems use a neural network as the basis of the software. This is a software encapsulation of a few neurons, also emulated in software and known as perceptrons. Just like our neurons, perceptrons receive stimuli in the form of input signals, and fire off a single signal as output, provided the sum of (or the mix of) input signals is greater than some value. Neural networks need to be trained. In other words, they aren't written as fully functional for a problem space – they have to be taught. The programmer has to feed many examples of features in the problem space to the neural network, observe the outputs, compare them with the desired outputs, then tweak the configuration of the perceptrons to make the output closer to the expected results.

The face detection 'algorithm' is one such neural network: Viola and Jones used a database of thousands of faces from the internet to tune their network to recognise faces accurately (according to their paper, they used 4,916 images of faces and 9,500 images of non-faces that they sliced and diced in various ways). The training took weeks.

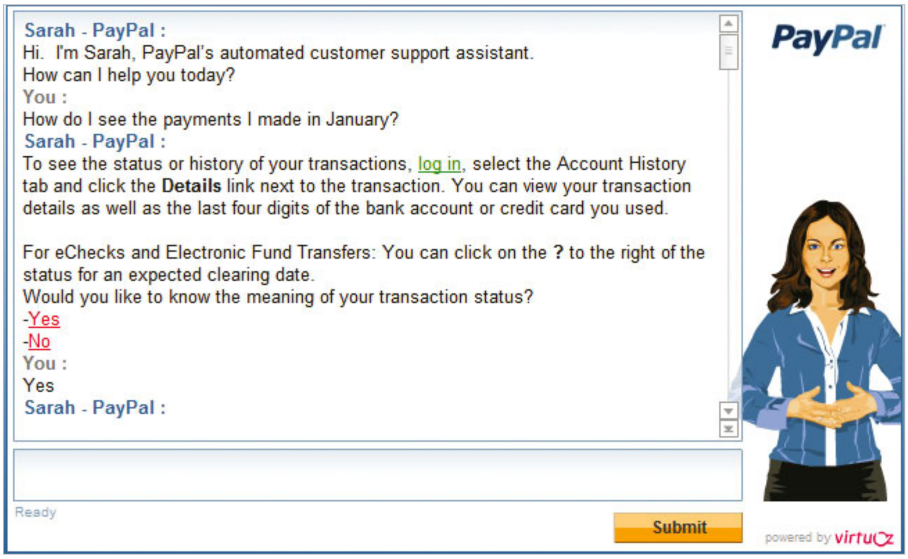
Another specific AI-like problem is OCR, or Optical Character Recognition. Again, the main engine is a neural network, and this time it's trained with lots of examples of printed characters from different fonts, and from high to low resolution. The success of OCR is such that all scanners come with it as part of their software, and the main reason for investing in commercial OCR packages is for slightly better recognition and the ability to fully replicate the formatting of the scanned document.

The kind of intelligence exposed by these types of programs is statistical in nature. We can see the success of these applications as embodiments of AI, despite the fact that they would never be able to participate in a Turing Test. Nevertheless, even as recently as a few years ago, such programs were inconceivable.

Such statistical intelligence isn't limited to image recognition or to translation engines. In 2008, Netflix promoted a prize asking for improvements on its movie recommendation algorithm. The winning algorithm (known as Pragmatic Chaos) uses a slew of factors to help provide recommendations, including temporal factors like movie popularity (charted over time), user biases and preferences as evinced by their changing tastes. In essence: using a lot of statistical data passed through various models to such an extent that the resulting system wasn't designed, but evolved.

As you've seen, we can view AI through two incompatible lenses. The first is what we've grown up with: a computer system and software that can emulate a human being to the extent that it can fool judges using the Turing Test. It's known as strong AI and is the subject of many science fiction movies, the most famous being HAL 9000 from 2001: A Space Odyssey. The second is perhaps more interesting, because it affects us in our daily lives: specific AI that solves single problems, and that would have been the subject of SF novels just a few years ago. What new specific AI awaits us in the next five years? **PCP**

Julian M Bucknall has worked for companies ranging from TurboPower to Microsoft and is now CTO for Developer Express.
feedback@pcplus.co.uk



Sarah - PayPal :
Hi. I'm Sarah, PayPal's automated customer support assistant.
How can I help you today?
You :
How do I see the payments I made in January?
Sarah - PayPal :
To see the status or history of your transactions, log in, select the Account History tab and click the **Details** link next to the transaction. You can view your transaction details as well as the last four digits of the bank account or credit card you used.

For eChecks and Electronic Fund Transfers: You can click on the **?** to the right of the status for an expected clearing date.
Would you like to know the meaning of your transaction status?
-Yes
-No
You :
Yes
Sarah - PayPal :

▲ **Figure 2: Sarah, the PayPal automated online assistant, is an example of a chatterbot.**

Natural Language

Computational Linguistics is the study of computers and how they can process human language. It's a fundamental part of strong AI, and of something that could pass the Turing Test.

Although a lot of research has been done in parsing human languages, identifying parts of speech, grammar and so on, the problem for strong AI is that understanding human speech seems to require an extraordinary amount of knowledge about the real world. For example, understanding the famous sentence "Time flies like an arrow, fruit flies like a banana" seems simple enough, but parsing it and identifying the parts of speech is a hard task that requires knowledge of time, flying, arrows, fruit flies and so on.

Nevertheless, we're used to word processors providing grammar and spellchecking hints, both of which require such automated parsing tools. Such tools will become better with time. ■