GRESHAM COLLEGE

28TH MAY 2019

CREATIVITY: CAN COMPUTERS CUT IT?

PROFESSOR RICHARD HARVEY FBCS

The word *creativity* is not very well defined, but in this lecture, I am interested in the uses of computers for what conventionally might be described as artistic pursuits: writing; painting; music; and so on. I am not going to be overly concerned with whether computers are really creative – until we have a better understanding of the human brain then we are not ready to answer that question[1] – instead we should be interested in how computers can create artistic works and what that tells about the process of creativity. A number of people feel rather threatened by the creative computer; creativity being seen as the last stronghold of human intellect. To those people I should point out that creativity exists from the mundane to the exquisite and even experienced critics will occasionally confuse the two. In this lecture you will have to exercise your own judgement as to whether the outputs are worthy or not.

That said, the standard test for creative competence in computers is the "Turing test". A Turing test was originally conceived as a test of artificial intelligence and is comprised of vote among humans between computer-generated and human-generated outputs. If the humans cannot tell the two apart then the test is passed. A number of scholars think the Turing test is rather problematic and ask if the vote take place among the general population or among a group of rarefied critics? Levels of artistic judgement are often quite low in the general population, yet the critics can get stuck in a "group think" that leads to homogeneity and bizarrely a suppression of originality[2]. That said, the Turing test is all we have so it is heavily invoked as the ultimate test of computer endeavour.

I'd like to start our exploration of creativity with a look at text. We discussed the technicalities of text processing in the previous lecture and I alluded to several systems that can analyse text. The current state-of-art uses deep neural networks but, before we get to those, I thought we just deviate into a couple of rather more rule-based systems in the field of computational humour. Humour is not very well defined so early workers had to focus on particular types of joke such as riddles and puns. One such example is The JAPE program dating from 1994. It searched through a large general-purpose lexicon known as WordNet looking for combinations that matched selected patterns (or *schemas* as they are called in the jargon). JAPE produced "jokes" like this:

What is the difference between a pretty glove and a silent cat? One is a cute mitten; the other is a mute kitten.

There may be a select group of people who find this rib-ticklingly funny, but for most of us I suspect, it raises merely a weak smile. Nevertheless, tests with humans proved that these were funnier than non-jokes but not as funny as human-constructed jokes[3]. In the next phase, in a system called STANDUP, the rules were increased in their sophistication and the system was used as a teaching aid for cognitively impaired children. Despite these systems, by the early 2000s there was a sense of despondency about computer generation of text — it was all very difficult and the rule-based approach, while very instructive and analytical, looked arcane and unscalable to real

[1] "Are computers creative?" is analogous to the question "Can machines think?" and the famous Computer Scientist Edsger Dijkstra is maybe best known for stating "The question of whether computers can think is not more interesting than the question of whether a submarine can swim."

[2] For example, many super-fans of the composer Benjamin Britten feel his music was sung best by Sir Peter Pears. Others, me included, feel it was ruined by Pears.

[3] As an aside I should note that it is commonplace for people ignorant of computer science to be loftily scathing about early-stage work like this. I've never understood this – the same thing does not happen in Physics or Chemistry — could it be that insecure people feel threatened by automation?

problems. All that changed with the introduction of deep-learning and some cool tricks to map from text into numbers.

To see that new approach in action, let's start with journalism. Journalism is an interesting domain as the opportunities for creativity are somewhat restricted — the facts are often fairly incontestable, and it is the role of the journalist to present them in a digestible fashion. Consider the following prose:

The Chargers take down the Chiefs
Even with an unexceptional outing for Philip Rivers, the Chargers handled the Chiefs, 37-20, at Arrowhead Stadium.
Rivers found the end zone for two touchdowns against the Chiefs on 18 of 23 passing for 209 yards and one pick. Matt Cassel went 24 of 42 with 251 yards passing, two touchdowns and three picks for the Chiefs.
…
Lastly, Nick Novak was perfect, hitting all three of the field goals he attempted…

For British readers the style is somewhat alien, partly because the patois of American Football is unfamiliar to us and partly because US journalistic practice varies. Nevertheless, it appears to be an acceptable account of an American football game. Certainly, human readers agree, and when compared to an account written to a Sports journalist, many readers preferred this version, which was written by software developed by Automated Insights. When confronted with such outputs it is frequent for professional writers to exclaim that such a thing is merely a stunt, a game or a computational whimsy. An automatic generator could never produce a sonnet, they claim. Here is an automatically produced sonnet:

The pass through the mountains
To step away, and lands in front of me
The tunnel, and I slow into a smile,
And when, I see that possibility
A moment of support it takes a while.
A sort of lyric is the little hand
To try if I can let it understand

To be fair I should note that the programmers created the title non-automatically [1] but I should also note that this system was the winner of the 2018 Poetry Turing Test, Poetix. The rules of Poetix state that the system should be able receive a prompt and contract a sonnet around the prompt. The implication of winning the competition is that there is certainly a large subset of humankind who cannot tell this apart from the genuine item.

The technology behind these innovations is irritatingly opaque. In the case of Automated Insights there is the usual commercial reluctance to reveal how their system works and in the case of the sonnet generator [1] the paper appears to be a collection of fixes applied to the basic technology which is a type of deep neural network known as an LSTM (long short-term memory network) plus a technique known as *word embedding*. Word embedding was covered in previous lectures and is a way of converting words into numbers. A naïve way of converting words into numbers would be to use a "one hot" code. If we had a language where the vocabulary consisted of the word's "cat", "mat"' "on", "sat" then one might represent these as [1000], [0100], [0010], [0001]. This is quite convenient for machine learning as we can imagine one output per word. This technique does work for letters of the alphabet and, in the slides for lecture I have an example, where this is used to re-create speeches of President Trump. However, for 10,000-word vocabularies we need some compression (10,000 element input vectors are unpleasantly large). The standard technique, known as Word2vec, was covered in previous lectures. It creates an "information bottleneck" by forcing a neural network to learn to replicate the input but only allowing 300 or so connections to do so. This abbreviated and approximate vector is then fed into the LSTM which is a special type of neural network in which the output is fed back into the input — this feedback gives the network a memory of what it has seen before [2]. Such systems, when extensively trained on, say, the speeches of President Trump, can be fed random noise as input and will output Trumpian language. Or they can be fed a word of phrase and will extrapolate from that.

When it comes to the visual arts, machine learning is on safer ground — images are very conveniently represented by sets of numbers called pixels as we saw in previous lectures. When one thinks of creativity, the idealised image is a painter. Creating painterly renditions from photographs has a noble history and is usually known as non-photorealistic rendering[4] or NPR. Along with a colleague Andrew Bangham, I worked on early NPR system in which we used a scale-space processor called a *sieve* (see previous lectures) to decompose the image into regions. By throwing away some detail the digital artist could reconstruct the image to create painterly pictures. This mode of operation was the default developed in the twentieth century: artists assistants or libraries of tools that allowed human artists to increase their repertoire or create novel effects and impressions. However, a fair-minded person would have trouble in ascribing creativity to such a system — the human artist was still in control and took the decisions about which tool to include.

The next phase of development was to build systems that could copy the style of a human artists. Early attempts amounted to finding a set of rules by which artists worked — the classic example being pointillism filters which replaced pixels with small circles of colour to emulate the works of Seurat and Signac. Such rule-based systems were not very convincing. However, the current state-of-the-art [3] makes ingenious use of a deep neural network, the VGG system, that was originally designed to recognise objects. The authors noted that as information progresses through the network, each layer provides information that might be adapted depending on their artistic style. So, if we pass into two VGG networks a photograph and a painting we can mix the two hierarchies together to a convincing impression of someone painting a novel scene in the manner of particular artist. However, a sceptic would again claim that much of the originality came from the original photograph and, furthermore, the digital human artist has exercised creativity in the selection of the appropriate painterly style. For true creativity we need an original subject in an original style.

For that we turn to "A Portrait of Edmund Bellamy" [4]. Obvious Art, a collective of artists based in Paris, created an algorithm that creates original pictures using a unique style[5]. This painting was auctioned at Christie's in New York in 2018 and raised $435,000 dollars against an estimate of $10,000. The system they use is called a generative adversarial network or (GAN). The thinking behind a GAN is to firstly develop a system called a "discriminator" which it is hoped will be able to distinguish between real and fake art. Such a system can be trained by feeding it many portraits. To make fake art, random noise is fed into another deep neural network. In the first few million iterations the fake art looks rubbish and the discriminator has an easy job. But as the networks learns more, the discriminator and the generator become more discerning. And after many many days of training the point is reached where occasionally the generator fools the discriminator and now, we have art. It's random noise that has been processed by a network to produce "fake" art that cannot be differentiated from real art.

GANs are powerful[6] an have also been used to create photo-realistic human faces. And the latest generation of GANs are able to separate out large and small-scale variations so the, say, the pose and ethnicity of one person, could be applied to another[7].

When it comes to music there is a pressing commercial imperative. Music is ubiquitous and music rights are aggressively pursued by musicians. So, were automatic construction of music to order possible, then there would be money it. Even the relatively niche task of musicology can, in certain domains have revenue opportunities[8]. In classical composition the current state-of-the-art is Iamus or, its commercial variant Melomics. These systems create musical scores which have recently been recorded[9] by human players. As the system has commercial aspects, the true algorithm behind Melomics is not published. However, it would appear to be based around what is known as a *genetic algorithm*. Genetic Algorihms (or Gas) are a class of algorithms that mimic human evolution. The first aspect of the algorithm is coding the problem into a string of binary digits. This is the genetic code. Each

[4] It's a rather convoluted phrase that is meant to emphasise that much of computer graphics aims to produce synthetic images, or renderings, that fool the eye into believe it is looking at reality (photo-realistic rendering).
[5] Although as I am based at the University of East Anglia which has, at the Sainsbury Centre for Visual Art, an unrivalled collection of Francis Bacon portraits, I find myself being drawn to parallels with Bacon's paintings.
[6] But tricky to train as one has to train both the generator and discriminator simultaneously. It is not unusual for GANs to fail to train.
[7] In fact the authors go further and take the style of a car and apply it to someone's bedroom — one feels that there is a new business in that idea alone!
[8] Pop music companies usually employ an "A & R" department who essentially talent spotters. This function has recently been automated via Music X-ray.
[9] The computer's first album was called "Iamus" and was recorded by the London Symphony Orchestra in 2012.

composition, or part of a composition, would have a unique code. Next the programmers try to define what they mean by a "good" composition. In the parlance of GAs this is called "fitness". The algorithms are usually initiated with many, many, random codes. These compositions sound mostly awful of course but by evaluating the fitness we can determine the best sounding ones. At this point algorithms vary, but typically at this stage we might kill off the unfit algorithms and breed the fit ones with each other to create a modified population that after many many million of iterations creates better compositions.

Creativity therefore has been revolutionised by the current technology of deep learning or deep neural networks. Of course, architectures are still evolving but the current thinking is, if it is possible to define what is meant as good art (or real art versus fake art) then time and many CPU cycles later, art will emerge. In this sense, computer artists are much like some real artists they are not very insightful about how or why they create art, but they have a finely developed sense of what they like. My position therefore is essentially optimistic about the progress and prospects of computer creativity. However, I should add one caveat which is that we have more discussed creativity in what I regard as its purist sense — invention and problem solving. These seem to me quite different and require a whole new lecture.

For those who prefer a more dystopian vision, you may wish to recall the following quote from George Orwell's *Nineteen-eighty-Four*,

"And the Records Department, after all, was itself only a single branch of the Ministry of Truth, whose primary job was not to reconstruct the past but to supply the citizens of Oceania with newspapers, films, textbooks, telescreen programs, plays, novels – with every conceivable kind of information, instruction, or entertainment, from a statue to a slogan, from a lyric poem to a biological treatise, and from a child's spelling-book to a Newspeak dictionary. And the Ministry had not only to supply the multifarious needs of the party, but also to repeat the whole operation at a lower level for the benefit of the proletariat. There was a whole chain of separate departments dealing with proletarian literature, music, drama, and entertainment generally. Here were produced rubbishy newspapers containing almost nothing except sport, crime and astrology, sensational five-cent novelettes, films oozing with sex, and sentimental songs which were composed entirely by mechanical means on a special kind of kaleidoscope known as a versification."

This leads to a number of interesting questions relating to technology and its interaction with society. Those questions are very timely, which is why my next series of lectures on The Digital State will look at the role of computer in solving problems in society and we will examine if, like Orwell's versificator, technology has an unequal impact on society.

1. "Shall I compare thee to a machine written sonnet", John Benhart, Tianjin Duan, Peter Hase, Liuyi Zhu, Cynthia Rudin, arXiv:1811.05067, Nov 2018
2. "Recurrent neural networks and LSTM tutorial in Python and TensorFlow", Andy Thomas, https://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/
3. "A neural algorithm of artistic style," Leon A Gatys, Alexander S Ecker, Matthias Bethge, arXiv:1508.0657, Aug 2015.
4. https://obvious-art.com/edmond-de-belamy.html