An Architecture for an Expressive Responsive Machine

Laura Dekker
Computing Department
Goldsmiths, University of London, UK
laura@theretohere.co.uk

This paper describes work in progress, an architecture for an expressive machine, which learns, senses and responds to its environment with 'creative' output. This is not to mimic a human or any other known biological organism, but an attempt to investigate what it might mean for a machine to do this 'on its own terms'. This hardware and software system forms the functional core of a large-scale interactive art installation, which plays with transduction between the material and non-material worlds, and between signals/stimuli in multiple forms. The meaning of 'expression' is discussed in reference to the machine. Some recent artworks from other artists are briefly reviewed, artworks which also employ a machine's-eye view of the world. I discuss what expression might mean for these machines, and who or what might be the intended audience. Following from the artistic impetus for my work, the design rationale for the machine is presented. A modular architecture, with asynchronous messaging, allows for experimentation with various methods of pattern recognition, sensing and activation. Some sensing and expression modes are described in more detail. A neural network is trained to process live video input stream, to 'taste' what the machine is seeing in the world. A Responsive Markov Model is developed, with an automated method to extract vocabulary from selected sources (amongst others, Shakespeare's sonnets and Molly Bloom's ending monologue in James Joyce's *Ulysses*) and the ability to respond to particular stimuli, in real time, in generating new texts.

Art installation. Neural networks. Markov Model. Machine creativity. Intelligent agents. Machine learning.

1. INTRODUCTION

In the early 1950s, Christopher Strachey, working alongside Alan Turing on the Manchester University Computer, programmed it to create 'machinic' love letters (Link 2009, Goriunova 2014), which he displayed on the lab walls:

SWEETHEART MOPPET

YOU ARE MY ANXIOUS LONGING: MY
ANXIOUS LOVE. MY SYMPATHETIC HEART
FERVENTLY YEARNs FOR YOUR
SYMPATHY. MY WINNING ENTHUSIASM
LOVES YOUR PASSION. MY DEAR
ENTHUSIASM BURNINGLY TEMPTS YOUR
ENCHANTMENT.

YOURS AVIDLY

M. U. C.

Figure 1: Some Output of the Love Letter Generator

My art practice is concerned with the reciprocal relationships between humans and technology, in making and coping with the world. Following from

Strachey, this paper describes my work in progress, to build an expressive machine, embodied in hardware and software, which learns, senses and responds to its environment. The emphasis is on that part of artificial mind involved in sensing, processing external stimuli into internal states (loosely related to human states of emotion, stimulation, etc.), and then expressing those internal states via output, such as poetry and audioscapes. For the moment we put aside considerations of whether or not the machine here is to be considered truly 'creative' (though it might be – for example, it just about meets Boden's requirements of *combination*, *exploration* and *transformation* (Boden 2004)). Of more concern here is 'expression', which is used here more or less in its ordinary dictionary definition as 'showing how you feel'. This is emphatically not to mimic a human or any other known biological organism, but an attempt to investigate what it might mean for a machine to do this 'on its own terms', to have its own aesthetics and drives, if such a thing is possible. Later I unpick and discuss in more detail

about what expression means in reference to the machine, and the algorithmic issues that arise.

The system described in this paper forms the functional core of a large-scale interactive art installation, called *Salty Bitter Sweet*, which has been exhibited in several instantiations (Dekker 2017). In the implementation described here (figures 2 and 3) the machine presents itself as an assemblage of raw, exposed parts, held awkwardly together on lab retort stands, clasped by rubbery three-pronged clamps. A webcam with torn casing sweeps slowly, rotating, along a suspended track, seeking out things of interest. Below is a pile of junk: decaying and live organic matter, semi-functioning and dormant gadgets. Wires and supports go down into, or arise up from this pile. Where the machine begins and ends is not clear.

Figure 2: *Salty Bitter Sweet*: webcam, perspex 'inner eye', microcontrollers, Arduinos, thermal printer, junk.

Figure 3: *Salty Bitter Sweet*: projected output of generated text and internal states.

A large translucent disc, suspended above the stands, clamps and electronic components, glows with a moving image. On further observation, it seems to show the view that the camera must be seeing right now. A chunked pixelated centre indicates, perhaps, an area of more concentrated focus. Below the junk-piled surface, some projected moving bars – meters of some sort – indicate 'salty, bitter, sweet, sour, umami'. Next to this, some text: sentences that gradually change, update. Now and then a small thermal printer buzzes, advancing a narrow roll of paper, on which is printed some text.

The machine is firmly rooted in the materiality of the world, not just in the sense of its own physical embodiment in hardware, but in its response to matter, transforming the decaying technological and organic detritus it experiences into its own expressive output.

2. BACKGROUND AND RELATED WORK

2.1 Beyond the human

This work takes place from a movement of anti-anthropocentrism across a range of disciplines, including computer science, the arts, social sciences and philosophy: reconsidering what it is to be human and exploring the agency of nonhuman entities.

Haraway's 'A Cyborg Manifesto' (Haraway 1991) uses the concept of the cyborg metaphorically, to reject rigid boundaries in gender and sexuality, the privileging of human over animal, and the boundaries between human and others. This is not just to remove those constraints, but also the identity politics that arises as a reaction to, and arguably reinforces them. Braidotti (2013) expands on the debate, proposing that the 'posthuman move' is an opportunity to 'empower the pursuit of alternative schemes of thought, knowledge and self-representation'; to think critically and creatively, in order to develop a more sustainable relationship with the planet and its other inhabitants.

Object-oriented ontology further develops this anti-anthropocentric thought, to consider inanimate objects' existence and experience, away from human perception and interaction (Bogost 2012, Harman 2014). Goriunova (2014), Parisi & Fazi (2014) and others pose questions about the machine's experience, attempting to consider, in a rigorous way, such things as 'do machine's have fun?' and what exactly 'fun' might mean for a computer. Fabulation and fiction have long been used as a vehicle for speculation of this kind, for example, in Fuller's recent essay written from the

machine's sensually rich point of view (Fuller 2017).

2.2 Unpicking expression

The manifestation of 'expression' in Strachey's *Love Letter Generator* program is a continuous and uninhibited flow – a kind of 'stream of (un)consciousness'. Every cycle it manipulates a tagged sentence structure, substituting words with random selections from a restricted vocabulary set. It seems to be an expression of the machine's state of love for... for whom, or for what? Another machine, perhaps? The object of this attention – the implied audience – is not clear, although it is presented in words of the English language, in human readable form, so perhaps it is for a human after all.

In general – machine or otherwise – expression can be considered as the transduction of an internal state into a form that can be externalised, and then presenting that externally. In reference to the machine, the algorithmic issues are:

- Triggering conditions.
- Internal states for expression.
- Temporal (e.g. delay, speed, flow).
- Immediate external input/stimuli.
- Other stored information, knowledge.
- Synthesis/transformation algorithms.
- Choice of output modes, with possible consideration of a target audience.
- Output/presentation 'devices'.
- End conditions.

This does not necessarily require an audience waiting in synchrony, although the choice of output mode *might* take into consideration the capabilities and disposition of a possible audience, which might, but not necessarily, be human.

2.3 Art of machine expression

Creating a system that expresses itself 'mechanically', and yet produces behaviour or other output that we can begin to make sense of, seems that it might be, by definition, an impossible task. But that is the kind of challenge that is well suited to art and artists, making use of advances in sensors, machine vision and machine learning.

Patrick Tresset's drawing robot, *Paul*, which observes and responds to the world, has been developed as a performative art installation (Tresset & Fol Leymarie 2013). The machine carries out observational face drawings of sitters (usually human), using its own style that arises out of the material execution of its algorithms, its physical actuators and drawing tools. Tresset describes his machines as 'robotic agents as actors

which are evocations of humanness', with 'expressive and obsessive' aspects to their behaviour. The machine's expression is triggered by the presence of a human sitter, and executed via the drawing algorithms, including a method for deciding on the drawing's 'completeness'.

Nye Thompson's *The Seeker* is a 'machine entity that travels the world virtually, and describes for us what it sees' (Thompson 2017). It reveals initially endearing virtuous errors in its interpretation, which point to the serious business of the inescapable biases in how technology interacts with, *and makes*, the world. Expression here seems to be in terms of the machine's drive to explore and interpret, and its preoccupation with, or anxiety about, a particular world view.

Figure 4: © Nye Thompson, 'The Seeker: Visions', 2017.

'Away from the human' is deeply explored in Glitch Art. Briz (2011) puts forward a philosophy of Glitch, to raise awareness and establish a critical relationship between human (i.e., *users*) and machine. In his 'tactful exploitation of systems', the human acts as facilitator for the machine's expression, by intervening at the code level. Briz guides the layperson through processes to examine the raw code of digital video and make systematic disruptions to generate unpredictable (at least for the human) artefacts. He proposes that these artefacts perhaps express a 'natural' machine aesthetic.

In the *Google-Assisted Living* gallery of Mark Amerika's *Museum of Glitch Art, Lake Como Remix* (Amerika 2012) records a journey via a corrupted Google Streetview's model of the world. It eerily follows a flattened motorbike along the shores of Lake Como and through mountain tunnels. Impossible planes helpfully support impossibly stretched pixels in an attempt to reconcile irreconcilable surfaces; sticky strands of stretched pixels attempt to span unexpected gaps; a sudden

onset of flat blank greyness fills in for the unknown. The occasional small artefacts of a *normal* experience of Streetview expand as the main content. The machine is taking you on its own journey, demanding that you see the world from its own point of view.

Helen Pritchard's *Critter Compiler* is a 'microbial prototype novel writer and a speculative experiment' (Pritchard 2016). Though there is a much broader context and implications of this work, here I home in on the system core of the piece. A computer runs some code – a neural network, which has been trained at the character-based level, using George Eliot's *Middlemarch*, in order to write its own novella. In executing the code, the processor also generates heat, which encourages a contained colony of algae to proliferate. As they and their watery medium pass over the CPU, this in turn cools it, thereby allowing the CPU to continue its task of computation – the novel-writing process. The output is almost unintelligible for a human observer, but fragments of seeming intelligibility now and then leap satisfyingly from the screen.

Figure 5: *Critter Compiler Prototype* by Helen Pritchard, *Executions Exhibition*, Medea, Malmo (2016).
Image: Helen Pritchard.

BOB ('Bag of Beliefs') is part of Ian Cheng's recent *Emissaries* series, exploring emergent properties of artificial life (Cheng 2018). Several instances of BOB, each with a gently writhing caterpillar-dragon body, present themselves on screens, interacting with audience-participants via an iPhone – the collective BOBs' shared external sensory apparatus. Internal states record information, such as the inferred emotional state of a visitor, and BOB's own mental and bodily conditions. In determining his next move, BOB takes a 'snapshot' of his current states, and finds closest matches from memory. Continuous learning and several different 'mindsets' ('baby', 'child', 'adult', 'parent' BOB) add to the complexity of his behaviours. His modes of expression include sound, bodily transformation, movement and focus of attention – some apparently directed at his human observers, and some apparently just for himself.

3. SYSTEM DESIGN

3.1 Requirements

The main artistic requirements for my system are to allow freedom for experimentation, to encourage emergent behaviour, and to give a strong sensory presence (in material as well as behavioural terms) for the audience to experience. For that, the software/hardware needs to provide the following functionality:

- Range of sensing modes, including querying internal activity.
- Range of pattern recognition capabilities.
- Store and retrieve sensed data and encodings about the world.
- Ability to process and derive new states from existing states.
- Actuators to respond to stimuli, support sensing processes, and present output from expressive processing.
- Control focus of attention.
- Generate expressive output in various ways.
- Learning, to improve, correct and extend capabilities, and to create new capabilities.

By adopting a simple, modular structure, functions can easily be added, removed and adapted. Unlike many attempts at a unified model of an intelligent agent, e.g., du Castel (2015), SOAR (2017), here I want to remain agnostic, allowing for a heterogeneous mix, to take make opportunistic use of advances in pattern recognition and learning techniques. If modules communicate via messages with variable content (rather than a centralised control) then restrictions on their implementation and runtime conditions can be minimised:

- A module may be implemented in hardware, software or both.
- Run across one or multiple processors.
- Components can be modified and developed independently of each other.
- Modules can be nested.
- Multiple, changing, sensing modes and multiple output modes and devices.

3.2 Module organisation

Where possible these functions are separated out into independently running processes, which can be on the same processor or on another networked piece of hardware. Using the idea of 'hungry' operators (Dennett 1991) modules continually seek input, listening for messages and sending their output asynchronously. For example:

- A sensing process seeks input from the sensor devices that it knows about.

- A sensing process might make use of an actuator to obtain what information it needs (e.g., move camera).
- An internal state might modulate a sensing mode (e.g., to ignore a region it considers unimportant).
- An expressive process seeks information from the internal states.
- An activation process (e.g. expression) determines an internal state (e.g. 'satisfaction at creating').

Importantly, though much activity includes modules querying the internal states, it is not an actively coordinating centre – it is a loose repository of information of various types and complexity.

Figure 6: Overall system operation

4. IMPLEMENTATION

4.1 Tools and materials

4.1.1 Software libraries

C++ / openFrameworks0.9.4 / ofxOsc
Python 3.5 / BeautifulSoup 4 / Natural Language
Toolkit (NLTK) / Markovify / python-osc-1.6.3
Wekinator machine learning system

4.1.2 External data sources

Project Gutenberg (www.gutenberg.org)
No Fear Shakespeare (Sparknotes 2017)
BBC Weather (www.bbc.co.uk/weather)

4.1.3 Hardware

Macbook running MacOS 10.12.6
Arduino Uno R3 x 3
Logitech Pro C920 webcam
Servomotor, Mini V Linear Actuator
Mini thermal printer, projectors, audio speakers

4.1.4 Physical structures and material

Retort stands and clamps
Perspex panel for back projection.
Rubbish, etc., for the machine to explore.

4.2 Run cycle

Modules run as separate processes, across multiple networked processors, using Open Sound Control (OSC) for asynchronous messaging. Each module sends output to, and listens for input on, nominated ports.

During each execution cycle (set at approx. 30/sec) the machine's sensing modules poll their allocated device (including the machine's own internal structures). The input data are processed and stored in the internal states.

Each expression module tests for its trigger conditions, and if stimulated to do so, it generates its offering, making use of data from the internal states. It sends the result to the various output devices that it knows about.

4.3 Sensory processing and internal states

Three sensing modes have been implemented, seeking their input data with varying time cycles:

- Webcam frame capture.
- Weather data scraped from external website.
- Querying internal CPU load.

The webcam module reads frames at 30 frames/sec. It compresses them to a 10x10 chunked image, which is sent out via OSC message. Wekinator listens on the designated port for incoming data. A neural network in Wekinator has been trained to associate each frame with five taste values ('salty', 'bitter', 'sweet', 'sour', 'umami'), each ranging 0.0-1.0; these values are output via OSC message. The internal state repository listens for updates and stores the new values.

Each timeslot the % CPU load is polled; the value is converted to a float 0.0-1.0. The internal states repository listens for updates and stores the new value.

Every n minutes the weather sensing module sends a request to the BBC weather website (www.bbc.co.uk/weather) and picks up the local % likelihood of precipitation over the next hourly

period. This is converted to a float 0.0-1.0. The internal states repository listens for updates and stores the new value.

One derived state processor, 'molly', has been implemented. Each cycle, this module listens for input and calculates the new value:

$$\text{idleness} = 1 - \text{CPU_load}$$

$$\text{molly} = \text{idleness} * \text{salty}$$

Figure 7: Sensory processing

In the training phase of the video-to-taste module, a neural network in Wekinator is configured to take as input the compressed video frames. Each frame = 100 x RGB triplets (ranging 0-255), each labelled with five 'taste' values (ranging 0.0-1.0). 17 video clips were used as the training set. The clips include sweeps of piles of rubbish, rotting material, garden soil with fallen petals, compost heaps, and time-lapse video of decaying flowers. Each clip duration is approximately 30 seconds, at 30 frames/sec, giving a total of approximately 14,000 training frames.

4.4 Expression

One of the machine's main modes of expression in this implementation is to generate poetry. The poetry module is implemented in Python, and uses a Markov model, modifying and extending the Markovify library (Singer-Vine 2015) in several ways to form a new Responsive Markov subclass:

- Input text is automatically tagged with part-of-speech (PoS), to build a tagged chain.
- It develops an extended, structured vocabulary set, which it uses in the production decision process.
- It responds in real time to input stimuli in the production process.

Figure 8: Responsive Markov Model

4.4.1 Building the Markov chain

In this example, the Markov chain, which generates the sentence structures, is built from a selection of fifty of Shakespeare's sonnets in modern form (Sparknotes 2017). Given the URL of the website, the text is processed automatically. The Python script, with BeautifulSoup, scrapes from the website, locates and cleans the relevant sections. This is passed to NLTK's part-of-speech tagger, which uses the University of Pennsylvania Treebank tag set (Penn Treebank 2003). The cleaned and PoS-tagged output forms the input to the Markov chain generator.

4.4.2 Extended vocabulary

The extended vocabulary is constructed from a range of sources. One vocabulary set is scraped from a section of James Joyce's *Ulysses* (Joyce 1914) on Project Gutenberg. The last chapter

contains a section that is known for its almost complete lack of punctuation – a rich and earthy verbalisation of the stream of consciousness of one of the main characters, Molly Bloom. Using a Python script, the raw text is automatically scraped from Gutenberg, cleaned and PoS-tagged, then organised into the machine's hierarchical vocabulary structure. For each part of speech, words are organised by 'style', which can correspond to any kind of category, such as sentiment, voice, topic:

```
vocab_dict =  
{ 'PoS0': { 'style0': [word0, word1..],  
            'style_n': [word0, word1..],  
    'PoS1': { 'style0': [word0, word1..],  
            'style_n': [word0, word1..] }}
```

Words from the *Ulysses* source are tagged with the style 'molly', being distinctive of her choice of vocabulary. Other vocabulary styles fed to the system are: 'neutral', 'salty', 'bitter', 'sweet', 'sour', 'umami'. Those word sets have been selected for maximum ambiguity, to aid in the production of fluid sentence construction and open interpretation. The 'umami' words are extracted partly from Chirlian's *Programming in C++* (Chirlian 1990), words with specific technical purpose, but also having an everyday usage. Below are a few tagged words from the 'umami' vocabulary set:

```
buffer_VB matrix_NN bugs_NNS bits_NNS  
corrupt_JJ parity_NN byte_NN void_NN  
string_NN float_VB null_JJ stream_NN  
return_VB overloaded_VBN private_JJ base_JJ  
argument_NN declare_VB expression_NN  
static_JJ constant_JJ throw_VB exception_NN  
this_DT unsigned_JJ member_NN hex_NN  
cast_VB catch_VB virtual_JJ coax_VB  
union_NN
```

Similarly, the 'salty' words are ambiguous, in having the possibility of erotic interpretation, but equally having an everyday, more neutral, usage.

4.4.3 Output generation

Each operation cycle, the poetry module checks its trigger conditions, and if true, it consults the machine's internal state variables to obtain a set of normalised style values. During the decision process, these values determine the probabilities of choosing a word of the specified part-of-speech, from each style in the extended vocabulary set. The system uses a two-state decision process, with a random start token. Output of the poetry generation module is then sent to the display handler, to control how the new lines of text are presented via the projector/screen, and to a mini thermal printer. Some example output, triggered by the internal states 'salty', 'bitter', 'sweet', 'molly', 'umami', is shown in figure 9.

5. DISCUSSION

The modular system design has been helpful in facilitating artistic experimentation. For example, in an earlier version, the poetry generation module was implemented using a genetic algorithm, with the output display being a view into the development of its sentence generation. Because the interface between that module and the others with which it exchanges data (the internal states and the display functions) did not need to change, it was very straightforward to implement this new version employing a neural network.

By setting out the whole system architecture in this way, it should make further enhancements of any of the modules easy to implement. For example, the expression triggering is currently very simple (every n execution cycles), but it would be easy to extend this with more meaningful triggers, such as a threshold on the 'newness' of incoming patterns.

The poetry generated by the Responsive Markov Model has a pleasing 'machinic' awkwardness and surprise. Though it is written in English, and probably for a human audience (though perhaps its object might be another machine), it never attempts to pass for human poetry. It will be interesting to explore more deeply this edge between human interpretability and a more 'true' or 'native' means of expression for the machine.

```
From extended vocab, using: ['salty', 'bitter', 'sweet', 'molly', 'umami']  
Listening for messages on ('127.0.0.1', 12000)  
Without love, we would float a logic to remember you, in prime with your brutal!  
Gently nervously you are shimmering yourself out of the chasm.  
The private lips that over intensity overloaded your velvety cock, so their stri  
ng will deserve to lick overloaded with you.  
Your unused underworld will have to be proud of.  
Your unused softness will admit to love accidental of.  
Madly how, when you are the corrupt underworld in the loss but velvety to suppor  
t yourself or thrill your hex.  
But your softness bites like a soft byte, and shrivel your impatient charm.  
By only shimmering bits with yourself, you could toy, the tears of pain would bu  
ffer at the brutal chasm of her underworld.  
Gorgeously delicately you are shimmering your cherry vainly than pushing the wea  
lth delicately.  
Lc  
Tr  
t But your softness bites like a soft byte, a  
Fc and shrivel your impatient charm. w  
Yc  
ez  
You open sweating madness, why are you arousing all of your hex by shimmering it  
and keeping it to hex!  
But you, in lust with your feverish brutal qualities, that would thrill an ecsta
```

Figure 9: Output from the Responsive Markov Model.

Many areas of functionality have not been implemented, or only in rudimentary form, at the time of writing. The system does not currently have a feedback loop to make use of prior knowledge in controlling its focus of attention, e.g., to manage the actuators that determine the camera's view. In this version, learning is carried out as a separate phase; the machine has not yet implemented any capabilities for self-evaluation or for self-directed learning. This is the main priority for the next phase of development.

6. CONCLUSION AND FUTURE WORK

As discussed in the opening sections, this work has been driven by artistic goals to explore a machine's eye view of the world, and what 'expression' might mean in reference to the machine.

To investigate these ideas, a heterogeneous modular hardware and software system has been designed. The implementation to date illustrates a range of sensing modes, showing how they feed into the machine's internal states, loosely related to human emotion or mood. One sensing mode is described in more detail: a neural network trained to process a live video input stream, to 'taste' what it is seeing in the world, and to store the resulting data as an internal state. A simple example shows how a derived state can be generated from lower-level states. A poetry generation module illustrates how the machine can respond, expressing its internal states. An existing Markov model library is extended to create the Responsive Markov Model, which responds to stimuli.

The system is currently being developed to extend its pattern recognition and text generation capabilities to perform 'predictive' tealeaf reading. For this it will need more sophisticated control of the actuators involved in sensing, to use existing knowledge to influence the focus of attention. Additionally, the aim is to extend the system with means of self-evaluation and self-directed learning.

7. REFERENCES

- Boden, M. (2004) *The Creative Mind: Myths and Mechanisms*. Routledge, London.
- Bogost, I. (2012) *Alien Phenomenology*. University of Minnesota Press, Minneapolis.
- Braidotti, R. (2013) *The Posthuman*. Polity Press, Cambridge, UK.
- Briz, N. Glitch Codec Tutorial, Vimeo. <https://vimeo.com/23653867> (retrieved 3 March 2017).
- Cheng, I. (2018) In Conversation with Nora Khan and Ben Vickers. Serpentine Galleries, London, UK, 6 March 2018.
- Dekker, L. (2017) Salty Bitter Sweet, http://www.lauradekker.io/There_to_Here/salty_bitter_sweet.html (retrieved 3 March 2017).
- Dennett, D. (1991) *Consciousness Explained*. Little, Brown and Co., New York.
- du Castel, B. (2015) Pattern Activation/Recognition Theory of Mind. *Frontiers in Computational Neuroscience*, 15 July 2015.
- Fuller, M. (2017) Half 10k Top-Slice Plus Five. In Shaw, J. K. & Reeves-Everson, T. (eds). *Fiction As Method*, Sternberg Press, Berlin.
- Goriunova, O. (ed.) (2014) *Fun and Software: Exploring Pleasure, Paradox and Pain in Computing*. Bloomsbury Press, New York and London.
- Haraway, D. (1991) A Cyborg Manifesto. *Simians, Cyborgs, and Women: the Reinvention of Nature*. Routledge, New York.
- Harman, G. (2014) *Objects and the Arts*. Institute of Contemporary Arts, London. YouTube. <https://www.youtube.com/watch?v=QJ0GR9bf00g> (retrieved 20 October 2017).
- Joyce, J. (1914) *Ulysses*, Project Gutenberg. <http://www.gutenberg.org/ebooks/4300> (retrieved 1 November 2017).
- Link, D. (2009) LoveLetters_1.0. http://www.alpha60.de/art/love_letters (retrieved 1 June 2017).
- Open Sound Control. <http://opensoundcontrol.org> (retrieved 4 February 2017).
- Parisi, L. and Fazi, M. B. (2014) Do Algorithms Have Fun? On Completion, Indeterminacy and Autonomy in Computation. In Goriunova, O. (ed). *Fun and Software*, Bloomsbury Press, New York and London.
- Penn Treebank Project (2003). University of Pennsylvania. http://www.ling.upenn.edu/courses/Fall_2003/ling01/penn_treebank_pos.html (retrieved 12 October 2017).
- Pritchard, H. (2018) Critter Compiler. In Pritchard H., Snodgrass, E. & Tyžlik-Carver, M. (eds) *Data Browser 06: Executing Practices*. Open Humanities Press.
- Singer-Vine, J. (2015) Markovify. GitHub. <http://github.com/jsvine/markovify> (retrieved 1 November 2017).
- SOAR, University of Michigan, <https://soar.eecs.umich.edu> (retrieved 18 September 2017).
- Sparknotes. No Fear Shakespeare. <http://nfs.sparknotes.com/sonnets> (retrieved 7 November 2017).
- Thompson, N. (2017) The Seeker. <http://www.nyethompson.co.uk/the-seeker> (retrieved 27 February 2018).
- Tresset, P. and Fol Leymarie, F. (2013) Portrait Drawing by Paul the Robot, *Computers & Graphics* 37, pp. 348–363.