Looking Behind the Text-To-Be-Seen: Analysing Twitter Bots as Electronic Literature

Ville Matias Lampi

Master of Arts thesis

Visual Culture and Contemporary Art
Department of Art
Aalto University School of Arts, Design and Architecture
2017

Aalto University
School of Arts, Design
and Architecture

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi

Master of Arts thesis abstract

Author Ville Matias Lampi

Title of thesis Looking Behind the Text-To-Be-Seen: Analysing Twitter Bots as Electronic Literature

Department Department of Art

Degree programme Visual Culture and Contemporary Art

Year 2017 **Number of pages** 99 **Language** English

Abstract

This thesis focuses on showing how Twitter bots can be analysed from the viewpoint of electronic literature (e-lit) and how the analysis differs from evaluating other works of e-lit. Although formal research on electronic literature goes back some decades, there is still not much research discussing bots in particular. By examining historical and contemporary textual generators, seminal theories on reading and writing e-lit and botmakers' practical notes about their craft, this study attempts to build an understanding of the process of creating a bot and the essential characteristics related to different kinds of bots.

What makes the analysis of bots different from other textual generators is that the source code, which many theorists consider key in understanding works of e-lit, is rarely available for reading. This thesis proposes an alternative method for analysing bots, a framework for reverse-engineering the bot's text generation procedures. By comparing the bot's updates with one another, it is possible to notice the formulas and words repeated by the bot in order to better understand the authorial choices made in its design. The framework takes into account the special characteristics of different kinds of bots, focusing on grammar-based bots, which utilise fill-in-the-blank-type sentence structures to generate texts, and list-based bots, which methodically progress through large databases.

From a survey of contemporary bots and earlier works of electronic and procedural literature, it becomes evident that understanding programming code is not essential for either analysing or creating bots: it is more important to understand the mechanisms of combinatory text generation and the author's role in writing and curating the materials used. Bots and text generators also often raise questions of authorship. However, a review of their creation process makes it clear that human creativity is essential for the production of computer-generated texts. With bots, the writing of texts turns into a second-order creation, the writing of word lists, templates and rules, to generate the text-to-be-seen, the output for the reader to encounter.

Keywords Twitter bots, electronic literature, text generation, web poetry, procedural writing

Aalto-yliopisto
Taiteiden ja suunnittelun korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Taiteen maisterin opinnäytteen tiivistelmä

Tekijä Ville Matias Lampi

Työn nimi Looking Behind the Text-To-Be-Seen: Analysing Twitter Bots as Electronic Literature

Laitos Taiteen laitos

Koulutusohjelma Visual Culture and Contemporary Art

Vuosi 2017 **Sivumäärä** 99 **Kieli** Englanti

Tiivistelmä

Tämä opinnäytetyö keskittyy esittelemään, kuinka Twitter-botteja on mahdollista analysoida elektronisen kirjallisuuden näkökulmasta ja kuinka niiden analyysi poikkeaa muiden elektronisen kirjallisuuden teosten tutkimisesta. Vaikka elektronisen kirjallisuuden tutkimusta on tehty joitain vuosikymmeniä, ei erityisesti botteihin keskittyvää tutkimusta ole juurikaan tuotettu. Tässä tutkimuksessa analysoidaan historiallisia ja nykyaikaisia tekstigeneraattoreita, elektronisen kirjallisuuden tutkijoiden teorioita teosten lukemisesta ja luomisesta sekä botin tekijöiden käytännön huomioita bottien kirjoittamisesta, joiden pohjalta luodaan kuva botin luomisprosessista ja erilaisiin botteihin liittyvistä ominaispiirteistä.

Bottien lähdekoodi on harvoin vapaasti luettavissa, minkä vuoksi bottien analysointi eroaa merkittävästi muiden tekstigeneraattoreiden tutkimuksesta. Monet teoreetikot pitävät lähdekoodin lukemista olennaisena tapana analysoida elektronisen kirjallisuuden teoksia. Tämä opinnäytetyö esittelee vaihtoehtoisen tavan analysoida botteja. Botin tuottamien päivitysten vertailu keskenään auttaa näkemään botin lähdekoodissa käytetyt toistuvat kaavat sekä ymmärtämään tarkemmin botin tekstin tuottavia menetelmiä ja niihin liittyviä taiteellisia valintoja. Esitelty metodi ottaa huomioon erityyppisten bottien ominaispiirteet, keskittyen kaavapohjaisiin botteihin, jotka asettelevat yksittäisiä sanoja valmiisiin lausepohjiin, ja listapohjaisiin botteihin, jotka käyvät järjestelmällisesti läpi suuria tietokantoja.

Tutkimuksessa läpikäytyjen vanhempien elektronisen ja proseduraalisen kirjallisuuden teosten ja nykyaikaisten bottien analyysin pohjalta voidaan päätellä, ettei bottien analysoiminen tai tekeminen vaadi ohjelmakoodin ymmärtämistä: on tärkeämpää, että botin lukija/tekijä ymmärtää prosessipohjaisen tekstitaiteen lainalaisuuksia sekä tekijän valintojen merkityksen käytettyjen materiaalien kirjoittamisessa ja kuratoinnissa. Botit ja tekstigeneraattorit kyseenalaistavat usein myös tekijyyden käsitteen. Niiden luomisprosessien analyysi osoittaa kuitenkin kiistattomasti, että tietokoneavusteinen tekstintuottaminen vaatii ihmisen luovuutta suunnitteluvaiheessa. Bottien tekemisessä kirjoittaminen vaihtuu toisen asteen luomiseksi, sanalistojen, lausepohjien ja sääntöjen kirjoittamiseksi, joiden pohjalta botti tuottaa lukijalle näytettävät tekstit, joita kuvataan otsikon termillä “the text-to-be-seen”.

Avainsanat Twitter-botit, elektroninen kirjallisuus, tekstigeneraattorit, internet-runous, proseduraalinen kirjoittaminen

Table of Contents

Figures

1. Introduction

Electronic Literature at Large

Although the term 'electronic literature,' often abbreviated as 'e-lit,' carries in its name the legacy of the history of literature, as a field of study it is merely a few decades old. A lot of the most seminal research in the field starts with an exploration into the term itself and the variety of fields that converge in the production of works viewed as electronic literature. The terms 'electronic' and 'literature' are both often contested by scholars in the field: as it will be pointed out later, literary theory treats texts as unchanging artefacts of study, whereas electronic literature approaches them as processes working both in the production and reading of those artefacts. Furthermore, the term 'electronic', or 'digital' which is used by some scholars, is called into question both because the history of procedural generation of text is much longer than the history of computers and because computers are by no means necessary for this kind of text-production.

The Electronic Literature Organization (ELO), bringing together artists and scholars in the field "[t]o facilitate and promote writing, publishing and reading of literature in electronic media," defines electronic literature as "works with important literary aspects that take advantage of the capabilities and contexts provided by the stand-alone or networked computer."[1] Although stand-alone or networked computer can be taken to mean any digital device of today, the definition offered by Noah Wardrip-Fruin in his analysis "Five Elements of Digital Literature" expands this list to fit the needs of modern technology, "I mean literary work that requires digital computation performed by laptops, desktops, servers, cellphones, game consoles, interactive environment controllers, or any of the other computers that surround us."[2] Both of these definitions allow for computing to appear in any phase of the text-production: the capabilities of digital devices can be harnessed to create the works, digital devices might be needed to be able to read the works or digital technology can be used to publish and distribute the works.

Literature is generally thought to deal with text. When discussing the ELO definition of electronic literature, N. Katherine Hayles notes,

> The definition is also slightly tautological in that it assumes preexisting knowledge of what constitutes of 'important literary aspect.' Although tautology is usually regarded as cardinal sin by definition writers, in this case the tautology seems appropriate, for electronic literature comes on the scene after five hundred years of print literature (and, of course, even longer manuscript and oral traditions). Readers come to digital work with expectations formed by

---

1 "What is E-Lit?" Electronic Literature Organization, accessed October 7, 2016, http://eliterature.org/what-is-e-lit/.

2 Noah Wardrip-Fruin, "Five Elements of Digital Literature," in *Reading Moving Letters: Digital Literature in Research and Teaching*, eds. Peter Gendolla, Roberto Simanowski and Jörgen Schäfer (Bielefeld: Transcript, 2010), 29.

> print, including extensive and tacit knowledge of letter forms, print conventions, and print literary modes.[3]

By defining the field through the lens of literature, the works of electronic literature can be read with this literary history in mind. Wardrip-Fruin continues his definition by defining literature as "the arts that call our attention to language, present us with characters, unfold stories, and make us reflect on the structures and common practices of such activities."[4] While computation in digital devices can be used to produce a myriad of different kinds of artefacts, electronic literature is focused on the literary use of computers' processing capabilities. This literary use means the creation of artefacts with the readers' aesthetic experience in mind, "In what way are they texts? They produce verbal structures, for aesthetic effect. This makes them similar to other literary phenomena."[5]

From Text to Processes

Although all of today's print literature is also passed through a computer in the initial word-processing by the author, the layout and the computer-controlled printing presses, word-processed text or printed books are not considered electronic literature. Furthermore, even e-book versions of those printed books do not belong into the canon of electronic literature as they are "digital media artefact[s] authored almost completely by the arrangement of pre-created text and image data."[6] What separates even the e-book version of a traditional book from the works of e-lit is that its text is set by the author and the publisher and it is read in the same kind of straightforward sequence as any print book.

In his seminal book *Cybertext*, Espen J. Aarseth proposes another way of reading and defining non-traditional literature and introduces the concept of 'cybertext.'[7] His research "focuses on the mechanical organization of the text, by positing the intricacies of the medium as an integral part of the literary exchange."[8] Rather than focusing on the role of digital computing as a factor separating traditional literature from what he calls 'ergodic literature,' Aarseth points out that the difference lies in how the text functions and how the reader explores the text,

> During the cybertextual process, the user will have effectuated a semiotic sequence, and this selective movement is a work of physical construction that the various concepts of 'reading' do not account for. This phenomenon I call *ergodic* [...] If ergodic literature is to make sense as a concept, there must also be nonergodic literature, where the effort to traverse the text is trivial,

---

3 N. Katherine Hayles, *Electronic Literature: New Horizons for the Literary* (Notre Dame, IN: University of Notre Dame, 2008), 3–4. PDF e-book.

4 Wardrip-Fruin, "Five Elements," 29.

5 Espen J. Aarseth, *Cybertext: Perspectives on Ergodic Literature* (Baltimore: Johns Hopkins University Press, 1997), 3.

6 Noah Wardrip-Fruin, "Authoring Processes," *Grand Text Auto* (blog), January 23, 2008, accessed June 30, 2016, https://grandtextauto.soe.ucsc.edu/2008/01/23/ep-12-authoring-processes/.

7 He appropriates the term 'cybertext' from Norbert Wiener's book *Cybernetics: Control and Communication in the Animal and the Machine.* (Aarseth, *Cybertext*, 1.)

8 Aarseth, *Cybertext*, 1.

> with no extranoematic responsibilities placed on the reader except (for example) eye movement and the periodic or arbitrary turning of pages.[9]

Aarseth goes on to study a wealth of examples of cybertexts, ranging from *I Ching*, the traditional Chinese method of prophesising, to Julio Cortázar's *Hopscotch*, chatbot *ELIZA* and text-based MUD (Multi-User Dungeon) games, pointing out that the ergodic text requires its reader to perform more functions than just reading the text; an ergodic text can have different ways in which it directs its user[10] to explore and reveal different parts of the text.[11] Aarseth treats text as a "as a material machine, a device capable of manipulating itself as well as the reader"[12] and as "an object of verbal communication that is not simply one fixed sequence of letters, words, and sentences but one in which the words or sequence of words may differ from reading to reading because of the shape, conventions, or mechanisms of the text."[13] What is important in this kind of a text is not the resulting output of the textual machine but the procedures, mechanisms and conventions that create the text for the reader or with the reader.

Viewing text as a process is central to many theories of electronic literature. For example, Philippe Bootz's "functional point of view" of "computer literature" considers text to be "understood mainly as a process and not only as the object the reader sees on screen."[14] William Winder, introducing his theory of 'Robotic Poetics' (RP), notes, "Poetics is traditionally conceived as the science of message construction, a systematic study of texts that describes why and how they make sense or have a particular effect. Even in this traditional vein, poetics describes text generation, the ergonomics of texts and the aesthetics of combinations. RP pushes the notion one step further in that it studies what creates the mechanics of creativity."[15]

Types of Electronic Literature

So far it has been established that works of electronic literature are procedurally created for a literary and aesthetic effect. Although Aarseth points out that their processes need not be computerised, computers make it efficient to create procedures for a great range of different kinds of experiences, "It is the computer's ability to carry out processes of significant magnitude (at least in part during the time of audience experience) that enables digital media that create a wide variety of possible experiences, respond to context, evolve over time, and interact with audiences."[16] Due to the nearly unlimited capabilities of computers to create a variety of

---

9 Ibid., 1–2.

10 Aarseth prefers the term 'user' over 'reader.'

11 Aarseth, *Cybertext*, 74.

12 Ibid., 24.

13 Espen J. Aarseth, "Nonlinearity and Literary Theory," in *Hyper / Text / Theory*, ed. George P. Landow (Baltimore: Johns Hopkins University Press, 1994), 51.

14 Philippe Bootz, "The Functional Point of View: New Artistic Forms for Programmed Literary Works," *Leonardo* 32, no. 4 (1999): 307, doi: 10.1162/002409499553307.

15 William Winder, "Robotic Poetics," in *A Companion to Digital Humanities*, eds. Susan Schreibman, Ray Siemens and John Unsworth (Oxford: Blackwell, 2004), accessed May 12, 2016, http://digitalhumanities.org/companion/view?docId=blackwell/9781405103213/9781405103213.xml&chunk.id=ss1-4-11.

16 Wardrip-Fruin, "Authoring Processes."

experiences through different procedures, expanding with the constant development of new technologies, works of electronic literature can appear in many different forms.

Leaving out Aarseth's categorisation of ergodic literature, Bootz, in his article published already in 1999, provides a handy classification for different works of electronic literature which is still applicable to the variety of works of e-lit created today, "All these texts can be traced back to three basic forms: hypertext, automatic generators and animated texts."[17] In his article, Bootz outlines the European developments in e-lit, noting that during the early years of computerised literature, 1978–1985, hypertext was the main focus in the United States, while European authors were chiefly interested in automatic generators.[18] Hypertext narratives or hypertext fiction is characterised by the use of hyperlinks pointing to different fragments of the text. The reader is free to choose their own sequence from the choices offered by the links, making the text read by one reader very likely different from the reading experience of another reader of the same narrative. The best-known examples from the 'Golden Era' of hypertext fiction are Michael Joyce's *afternoon, a story* and Shelley Jackson's *Patchwork Girl*, both published by EastGate Systems in the first half of 1990s on their proprietary hypertext reading and authoring system StorySpace.[19]

Bootz notes about automatic generators, "The automatic generator, based on the generative grammar of Chomsky, creates texts from a dictionary and grammar tools. Texts generated by computer are not written by the author of the program."[20] More generally, automatic generators are procedures by which words from word lists or dictionaries are placed into 'fill-in-the-blank'-type placeholders in formulas that can be fairly simple or very complex based on the author's intentions. The procedures of choosing the words can, per the immense possibilities offered by computers, either be random or based on, for example, user interaction, environmental variables or data streams. Another way of automatically generating text commonly used in textual generators is a mathematical process called Markov chains. Generators that use Markov chain algorithms "work on arbitrary input, contrary to older syntheticly [sic] combinatory methods [...] which always process fixed, pre-inscribed words."[21] In methods based on Markov chains, the generator is given a source text—usually something written by humans—which the process translates into "transition probabilities tables which can be computed even without any semantic or grammatical natural language understanding."[22] The program does not understand the language in the source text but is, however, able to create new texts by placing together words that appear together in the source text. The success of the output text, be it measured by interestingness, grammatical correctness or sheer signal-to-noise ratio, is largely dependent on the amount of material in the source text and the statistical depth[23] of the initial analysis. In addition to being

---

17 Bootz, "The Functional Point of View," 307–308.

18 Ibid., 308.

19 Both of the works have just recently been made available on USB sticks to be read on computers running modern operating systems.

20 Bootz, "The Functional Point of View," 308.

21 Florian Cramer, *Words Made Flesh: Code, Culture, Imagination* (Rotterdam: Media Design Research, Piet Zwart Institute, Willem de Kooning Academy, Hogeschool Rotterdam, 2005), 75, accessed May 19, 2016, http://www.netzliteratur.net/cramer/wordsmadefleshpdf.pdf.

22 Ibid., 74.

23 Statistical depth means how the source text is sampled in the analysis, one pair of words or characters at a time or in larger clusters of words or characters. The deeper the analysis, the more grammatically correct the results will be. This, however, risks "having your generated text repeat large portions of the source text." (Allison Parrish, "N-Grams and Markov Chains,"

the method-of-choice for European e-lit authors in the 80s, generative art has maintained its popularity within the e-lit community throughout the years. N. Katherine Hayles notes this in her 2008 analysis of e-lit, "Generative art, whereby an algorithm is used either to generate texts according to a randomized scheme or to scramble and rearrange preexisting texts, is currently one of the most innovative and robust categories of electronic literature."[24]

Lastly, animated text intends to break the traditional habits of reading by "[introducing] time within the written text, as a part of it."[25] In Bootz's view, animated text "creates a continuum between text and not-text, a transition between texts."[26] Animated texts can take many forms as they can appear as video poetry, Flash animations or even web pages scripted to alter the text while being read.

The works that would be placed within these categories have certainly evolved from Bootz's time but many contemporary works of electronic literature still rely on the same principles as their earlier counterparts. Some might even say that, for example, hypertext fiction is going through a revival with a vibrant online community creating, sharing and reading works created with the open-source authoring system *Twine*. While the distribution might have become easier from the times of floppy discs and CD-ROMS, the basic tenets of hypertext fiction—links, forking narratives and user exploration—remain. In the same vein, automatic generators originally printing their output onto paper tape from computers like *Mark I* have evolved into working as automatic bots that push their output directly into social networking services. Computer games have also evolved from the text adventures discussed by Aarseth and can often still be read as electronic literature, although they do not necessarily fit well into Bootz's trichotomy.

Generative and Procedural Writing

This thesis will focus on automatic generators and procedural writing that make use of combinatory techniques, like the grammar-based approach outlined above. Combinatory text generation raises some essential questions both about nature of the reading experience and about the role of authorship in the creation of the texts. Procedural writing can produce a large number of texts, each different from another, using a set of materials that is much smaller in scale than the scale of the output.[27] Due to the sheer number of possible output texts, it is impossible for the reader to finish reading a work of text generator like one would finish reading a book. This raises the question of how a text with unlimited potential outputs ought to be read and analysed.

The difference between the rules that create the texts and the actual texts realised by running the program is also linked to questions of authorship, creativity and *poiesis*[28] in procedural

---

Decontextualize, accessed June 30, http://www.decontextualize.com/teaching/rwet/n-grams-and-markov-chains/.)

[24] Hayles, *Electronic Literature*, 18.

[25] Bootz, "The Functional Point of View," 308.

[26] Ibid.

[27] Wardrip-Fruin, "Five Elements," 33.

[28] "Writing native to the electronic environment is under continual construction (poiesis) by its creators and receivers. The neologism poietics engages this dynamism." (Stephanie Strickland, "Writing the Virtual: Eleven Dimensions of E-Poetry," *Leonardo Electronic Almanac* 14, no. 5–6 (Aug. & Sept. 2006), accessed May 19, 2016. http://www.leoalmanac.org/wp-

literature. William Winder identifies the different types of creativity in the creation of procedural systems,

> Generated text has a robotic author, itself created by a human programmer. There is a poetics of creating that author (here creativity lies in the writing of instructions); a poetics of generating a specific text (how those instructions play out to make a given text); and a poetics of reading generated literature (how the reader will read a text knowing, or not, that it is automatically generated).[29]

Similar questions are also raised by Bootz, "What is really a text? Can a text have no author? What does it mean to read a particular text amongst an infinity of possible texts?"[30] Instead of focusing on these questions of authorship and creativity in procedural artworks discussed by the authors cited above, writing their analyses in the 90s and the first decade of this century, this research will focus on a phenomenon that has had its second coming with the rise of social media networks—bots. Bots take procedural generation of text to a whole new level with their relentless pace of posting new textual samples directly to the forums where people do more and more of their daily reading, integrating into people's everyday experiences, instead of being something that a reader needs to actively seek out in order to read electronic literature.

What Is a Bot?

The *Cambridge Advanced Learner's Dictionary & Thesaurus* defines 'bot' as "a computer program that works automatically, especially one that searches for and finds information on the internet."[31] These small pieces of software running on internet-connected servers can be programmed to produce any repetitive task, including posting on social networks on their own. Mark Sample, a researcher and a professor of digital humanities well versed in electronic literature and bots, provides many examples of tasks performed by bots today,

> Bots are small automated programs that index websites, edit Wikipedia entries, spam users, scrape data from pages, launch denial of service attacks, and other assorted activities, both mundane and nefarious. On Twitter bots are mostly spam but occasionally, they're creative endeavors.[32]

---

content/uploads/2012/09/06Writing-the-Virtual-Eleven-Dimensions-of-E-Poetry-by-Stephanie-Strickland-Vol-14-No-5-6-September-2006-Leonardo-Electronic-Almanac.pdf.)

29 Winder, "Robotic Poetics."

30 Bootz, "The Functional Point of View," 308.

31 *Cambridge Advanced Learner's Dictionary & Thesaurus*, s.v. "bot," accessed October 11, 2016, http://dictionary.cambridge.org/dictionary/english/bot.

32 Mark Sample, "A Protest Bot Is a Bot So Specific You Can't Mistake It for Bullshit: A Call for Bots of Conviction," *Medium*, May 30, 2014, accessed October 2, 2016, https://medium.com/@samplereality/a-protest-bot-is-a-bot-so-specific-you-cant-mistake-it-for-bullshit-90fe10b7fbaa.

Darius Kazemi, an artist creating "bots and generators and other weird internet stuff,"[33] defines bots slightly more narrowly, "I think a bot– in my experience, a bot can live anywhere that there's technology for humans to talk to other humans. I think a robo dialer that calls you with a survey on the phone is a bot. It's just a bot that's happening on the telephone system instead of on Twitter."[34] This definition frames bots as pieces of software interacting and communicating with humans in the same contexts in which humans are used to communicating with one another. Historically, bots appeared as characters in the MUD adventure games available on the early internet and its predecessors.[35] Also the chatrooms on IRC were often populated by bots with whom online users could talk. Nowadays, bots appear in workplace software like Slack, tracking project progress or coordinating lunch choices, in messaging software like Facebook Messenger, selling flowers or clothes, and on corporate websites, handling customer service chats. These types of bots are all created with the intention of improving efficiency in business by automating mundane and repetitive tasks.

In her talk "Understanding Bots" at the ELO 2016 conference, another bot artist and educator Allison Parrish offered a more detailed description of the characteristics of creative social media bots. According to her, a bot is characterised by "repeated output / ... generated by a procedure / ... occurring over an extended period of time / ... embedded in an otherwise intention-typical context (i.e., non-procedural writing)."[36] Especially the last characteristic is essential to understanding how bots created with an aesthetic or polemical effect in mind differ from the other types of bots intended to automate mundane tasks: creative bots appear in contexts populated by texts that are non-creative and non-procedural, created mostly by humans.[37] Kate Compton, a PhD candidate of computer science and the creator of a popular generative grammar tool *Tracery* used to create bots, highlights the surprising nature of bots' timed posts, "I like to think of them almost as a cuckoo clock, that it's a little decorative object that you have in your life and you know that every hour or once a day or once a month, it's going to, I guess rather literally, tweet at you something new and bizarre and surprising."[38] When juxtaposed with the mundane, intention-typical context of social networks, posts by generative bots have the power to surprise the reader with aesthetic content while the reader might not be actively seeking an aesthetic experience. In addition to the non-creative–creative and intention–typical—intention-atypical juxtapositions created by embedding bot content within a user's timeline, Parrish also argues that bots are a folk practice that attempts to take back the public online space from commercialisation, much like skateboarding or graffiti attempt to reclaim the public space of the city, "[T]here are a lot of twitter bots whose only purpose is to use twitter as a surface for things that don't look like

---

[33] Darius Kazemi, "Tiny Subversions," Tiny Subversions, accessed October 11, 2016, http://tinysubversions.com/.

[34] Darius Kazemi, "I, Twitter Bot," interview by Ira Flatow, *Science Friday*, podcast audio, August 19, 2016, accessed August 25, 2016, http://www.sciencefriday.com/segments/i-twitter-bot/.

[35] Aarseth, *Cybertext*, 156–157.

[36] Allison Parrish, "Understanding Bots" (presentation at ELO 2016, University of Victoria, June 22, 2016), PDF presentation slides and notes, accessed June 19, 2016, https://s3.amazonaws.com/aparrish/procedure-vs-procedure-elo2016.pdf.

[37] Although there are a lot of spam bots on Twitter, for example, it is hard to imagine a user following spammers, which, thus, keeps the timeline browsed by the user relatively human.

[38] Kate Compton, "I, Twitter Bot," interview by Ira Flatow, *Science Friday*, podcast audio, August 19, 2016, accessed August 25, 2016, http://www.sciencefriday.com/segments/i-twitter-bot/.

tweets—reinscribing the public space with something that runs counter to Twitter's business plan."[39]

Bots as a genre of e-lit differ greatly from hypertexts and even other types of procedural text generators, which require the reader to actively access them. After subscribing to—or following, to use Twitter's terminology—a generative bot, the generated texts come to the reader instead. Furthermore, one of the affordances of social media is the shorter form factor—the 140-character limit on Twitter, for example—which lends itself better for convincing generative writing, as noted by bot artist and scholar of digital literature Chris Rodley, "The computer generated novels are, as they programmers freely admit, mostly unreadable. Sustained narrative remains a problem. Automated writing of the creative variety becomes much more convincing on a small scale."[40]

Creative bots come in too many forms to be exhaustively listed and categorised. Most are textual, posting, for example, narratives that fit into the 140-character limit set by Twitter, like *@MagicRealismBot*. However, there are also bots that posts visually striking combinations of emoji, like *@choochoobot* which posts trains and scenery procedurally created using emoji, or draw procedural images, like *@tinyneighbor* which posts images of procedurally generated houses. The code running them may utilise procedural grammars, like *@What_Hastings* that posts procedurally generated imaginative events happening in the city of Hastings, Markov chains, like *@StubbBot* which posts remixed tweets sourced from the Twitter archive of the former Finnish Minister of Finance Alexander Stubb, or may repurpose data streams into a human-readable format, like *@congressedits* which follows the stream of Wikipedia edits and highlights anonymous edits made to English Wikipedia pages from inside the US Congress. Some bots merely retweet tweets that fulfil characteristics set by the author, containing, for example, a certain grammatical structure, a rhyme scheme or specific words, like *@prklsuomi* which reposts tweets containing Finnish curse words.

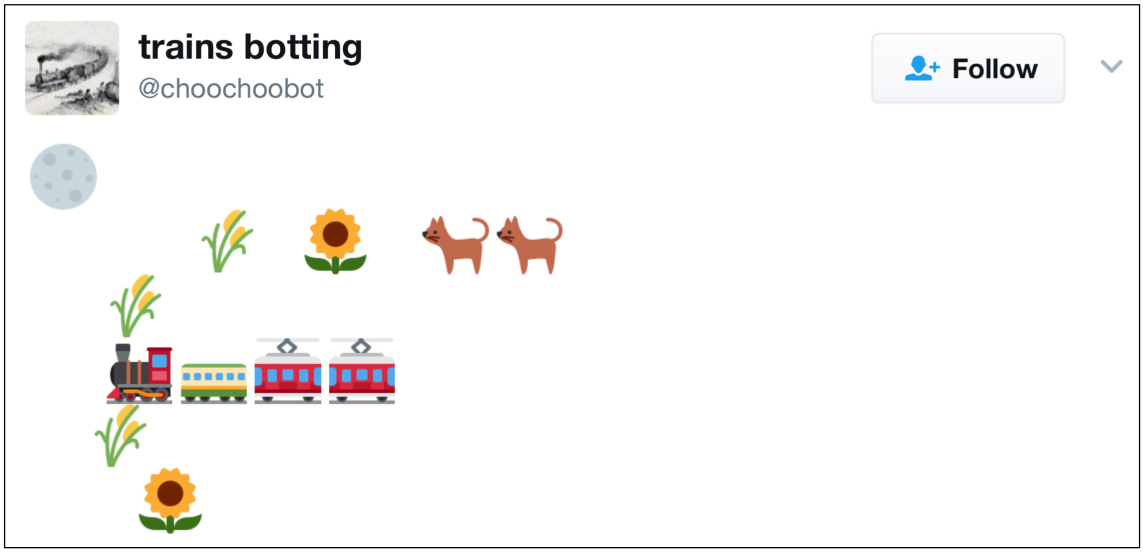

Figure 1. A screenshot of a @choochoobot *update, which shows emoji being used to create a miniature landscape. (https://twitter.com/choochoobot/status/858712808493432832, accessed May 2, 2017.)*

[39] Parrish, "Understanding Bots."

[40] Chris Rodley and Ali Rodley, "In Conversation with Chris and Ali Rodley: The Creators of the Magical Realism Bot," interview by Matthew Spencer, *Asymptote*, March 30, 2016, accessed October 2, 2016, http://www.asymptotejournal.com/blog/2016/03/30/in-conversation-with-chris-and-ali-rodley-the-creators-of-the-magical-realism-bot/.

The motivations behind each bot vary as well. They may be artistic or aesthetic in nature, characterised by, for example, "absurdism, comical juxtaposition, and an exhaustive sensibility."[41] They can also be journalistic in nature, exposing or visualising data in a way that humans can understand.[42] They may be polemical, exposing or protesting some issue in society.[43] All the different techniques may be used with all the different motivations, which makes the boundaries between different kinds of bots blurry and overlapping at best: a combinatory bot can be polemical, like *@thinkpiecebot* which posts made-up thinkpiece headlines about millennials "to call out the predictability of these articles,"[44] and an aesthetic bot can be driven by big data, like *@censusAmericans* which takes individual data points from the US Census data and translates them into human-readable narratives, telling short life stories of census respondents.

* * *

As shown above, the nature, timing and context of a reader's encounter with texts generated by bots differ from perusing other types of automatic generators and other genres of electronic literature. The aim of this research is to find out what the process of reading or encountering bots requires from the reader and how the earlier theories of electronic literature and earlier exemplary works of procedural writing can inform the analysis of bots as creative endeavours in the field of electronic literature. A categorisation of most common types of artistically-inclined bots will be developed and their shared characteristics discussed in order to develop a framework of aspects to consider to help the methodological reading and analysis of bots as electronic literature.

More specifically, chapter 2 will discuss early works of electronic literature and textual generation and chapter 3 different theories about the processes of creating and reading electronic literature. Chapter 4 introduces the different types of bots and the technologies running them in more detail. In chapter 5, the theories about earlier works of electronic literature will be applied to reading bots and the processes of creating and reading bots will be compared with other types of electronic literature. Chapter 6 will summarise the proposed framework for analysing bots and a few exemplary bots will be analysed in relation to the characteristics discovered in the research. Finally, chapter 7 will provide an outline of the findings.

---

41 Sample, "A Protest Bot."

42 Parrish, "Understanding Bots."

43 Ibid.

44 Nora Reed, "The Official @Thinkpiecebot FAQ," *Gusty Winds May Exist* (blog), September 6, 2015, accessed October 2, 2016, http://barrl.net/2748.

2. Historical Precedents

> One could believe that poetic computations of mobile data sets could not be imagined before modern computers were invented. However, the first modern art work based on a computational process and arbitrary input data dates back to 1923.[1]

In this quote, Florian Cramer is talking about Tristan Tzara's renowned performative method of creating Dadaist poetry by cutting up a newspaper article and randomly pulling out individual words from a bag filled with them.[2] Cramer notes that Tzara's method shares the characteristics of algorithmic processes utilised in modern computer artworks in that the method itself functions as "a simple computer creating random permutations of arbitrary input."[3] Tzara's cut-up poetry is one of many examples that scholars researching the roots of contemporary e-lit attribute as predecessors of the processes used in more modern procedural and generative works.

In his book *Words Made Flesh*, Cramer provides a survey of the history of executable code and computation in literary practices that precede digital computers by centuries. These practices range from the Kabbalistic and mediaeval practices of combinatory permutation on the names of God[4] and methods of organising encyclopaedias by combinatory categorisations[5] to Dada poetry and later literary experiments by groups like Oulipo (*Ouvroir de littérature potentielle*, workshop of potential literature) and writers like William S. Burroughs and Brion Gysin. Cramer is not alone in his analysis of the roots of modern combinatory work as analyses of the same literary references appear also in the histories of algorithmic procedures provided by Espen J. Aarseth, Noah Wardrip-Fruin and Loss Pequeño Glazier. This chapter will offer an overview of their analyses of the works that utilise algorithmic processes without the aid of a computer as well as of some of the first artistic works designed for the early computers. All of the works discussed here have a procedural nature and a lot of more literary examples of the predecessors of modern e-lit, like Nabokov's *Pale Fire* and Cortázar's *Hopscotch*,[6] have been left out as they are more related to the categories of hypertext and animated text mentioned in the previous chapter.

---

1 Cramer, *Words Made Flesh*, 75–76.

2 Tristan Tzara, "How to Make a Dadaist Poem," 1920, accessed October 13, 2016, http://www.writing.upenn.edu/~afilreis/88v/tzara.html.

3 Cramer, *Words Made Flesh*, 76.

4 Ibid., 29–32, 36–39.

5 Ibid., 39–41.

6 Aarseth, *Cybertext*, 7.

Literary Procedures

Cut-Ups Then and Now

As mentioned above, the creation of a Dadaist poem by following Tzara's procedure is a recombinant technique of text production, based on a fixed collection textual particles, or 'textons' to use Aarseth's terminology, from which the text presented to the reader is composed.[7] In the Dadaist poem, each word in the newspaper article is its own texton, leading to an increasing number of potential output texts when a longer source text is chosen for manipulation.

Tristan Tzara's procedure of manipulating ready-made texts inspired also other writers and artists, especially ones considered to be a part of the Beat generation. William S. Burroughs and Brion Gysin utilised the method extensively in their works and developed it further by both experimenting with randomising techniques in a large variety of media—in addition to text, they worked with tape recordings, film and even computer programs—and by developing their own method of the fold-in in which "page of one text folded down the middle on a page of another text."[8] For them, the cut-up method was a method of making the technique of collage, used by painters for fifty years prior, available to writers[9] and "to break the hold of the viral word and liberate resistances latent in language by freeing it from linear syntax and coherent narrative."[10]

In addition to text, Gysin worked extensively with sound recordings and edited his recordings utilising the cut-up methodology, turning conversations into randomised sentences and collections of sounds. Furthermore, recordings of sound poetry like "I Am That I Am," "In the Beginning Was the Word" and "No Poets Don't Own Words" utilise the cut-up method differently as they present the artist reading out different permutations of the same set of words over and over again. The method of producing these permutation poems also shows the close connection between electronic means of textual manipulation and the pre-electronic procedures as the original permutations read aloud by Gysin were created using a computer program,

> The poem ["I Am That I Am"] shuffles its words according to a formal algorithm. Its total of 720 permutations were calculated in the early 1960s, on a Honeywell computer with the aid of mathematician Ian Sommerville. Spoken by the author on a tape recording, this and other permutation poems of Gysin were not solely mathematical computations, but also incantations.[11]

While setting the language free from the bounds of grammar and syntax, the repetition of the different permutations of the same phrase does, indeed, give Gysin's poetry the sound of incantations, bringing to mind a search for a deeper meaning in randomised language and its broken structures.

---

7 Ibid., 60.

8 William S. Burroughs, "Note on Vaudeville Voices," in *The Moderns: An Anthology of New Writing in America*, ed. Leroi Jones (New York: Corinth Books, 1963), accessed October 15, 2016, http://www.writing.upenn.edu/~afilreis/88v/burroughs-cutup.html.

9 William S. Burroughs, "The Cut Up Method," in *The Moderns: An Anthology of New Writing in America*, ed. Leroi Jones (New York: Corinth Books, 1963), accessed October 15, 2016, http://www.writing.upenn.edu/~afilreis/88v/burroughs-cutup.html.

10 Hayles, *Electronic Literature*, 20.

11 Cramer, *Words Made Flesh*, 17.

Since Gysin and Burroughs, techniques that resemble the cut-up and fold-in methods have been widely used in a wealth of e-lit works. Techniques similar to the cut-up appear in works like Jim Andrews' *Stir Fry Texts*[12] and Geniwate and Brian Kim Stefans' *When You Reach Kyoto*[13], as noted by N. Katherine Hayles in *Electronic Literature.*[14] On the other hand, works like Loss Pequeño Glazier's *Mouseover*[15] exhibit functionality similar to the fold-in method, by combining larger chunks of text together with texts and images from other sources. The browser-based environment of these works enables the authors to make the combinatory process happen before the reader's eyes, at the reader's will, as mousing over the texts replaces parts of them with different ones, adding to the already dynamic nature of the cut-up method.

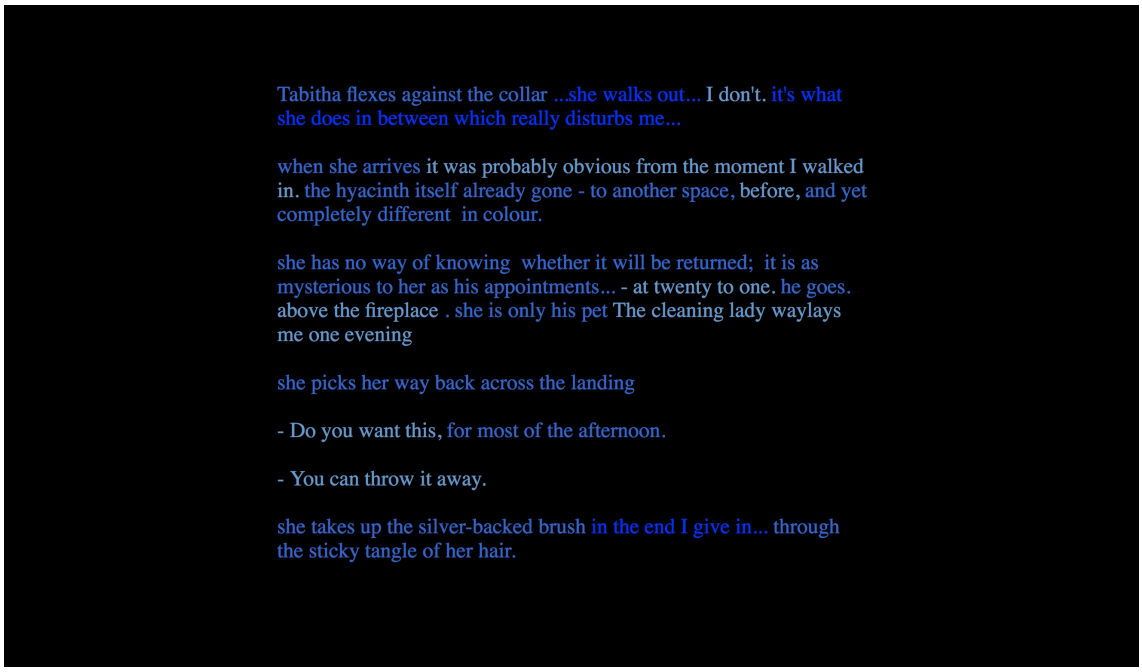

Figure 2. A screenshot of Jim Andrews' Stir Fry Texts*, which indicates different text fragments with different colours. (Accessed April 23, 2017.)*

Stephanie Strickland characterises these types of works by the term 'recombinant flux.' She also cites the aforementioned author Geniwate, noting the performative aspect of the cut-up strategy,

> Another recombinant literary system focused on message is Geniwate's visually sophisticated Concatenation [...] It addresses the situation of detention camps run by the Australian government. She cites the Oulipo and Burroughs's cut-up strategy as influencing her, but adds: 'Of course, it's not as random mas a cut-up; there are heaps of rules determining what gets

---

12 Jim Andrews, *Stir Fry Texts*, 1999–, in *Electronic Literature Collection* 1 (October 2006), accessed March 21, 2017, http://collection.eliterature.org/1/works/andrews__stir_fry_texts.html.

13 Geniwate and Brian Kim Stefans, *When You Reach Kyoto*, in *Electronic Literature Collection* 1 (October 2006), accessed March 21, 2017. http://collection.eliterature.org/1/works/geniwate__generative_poetry.html.

14 Hayles, *Electronic Literature*, 20.

15 Loss Pequeño Glazier, *Mouseover*, 1998, accessed April 23, 2017, http://wings.buffalo.edu/epc/authors/glazier/viz/mouseover/mouseover.html.

> generated. I'm not so much interested in the surreal aspect of the cut-up principle, but in the performative aspect.[16]

Indeed, by performing the combinatory process before the eyes of the reader, all of the aforementioned works of e-lit point out the performative aspect of the cut-ups: the reader facing the fluctuating text on the screen becomes acutely aware of the process that creates the syntax available for their reading. Furthermore, the combinations that occur while reading present the materiality of language and the fluctuating meanings attached to individual words and structures. This also occurs with Gysin's recordings as the reorganisation of the statements about one's existence, the world's existence and language shows how the meaning of a single phrase can shift when being modified ever so slightly. The performance of the combinatory process certainly releases the latent meaning in the source text, potentially pointing out, as Burroughs intended, that "[a]ll writing is in fact cut ups. A collage of words overheard."[17]

What is common in all these works is that the description of the process by which the texts are combined is an essential part of the work: The Dadaist method invites the reader to create their own poem, Gysin presents all the possible permutations in his voice recordings and the web-based works reshuffle texts right before the reader's eyes. In all of these cases, the reader is made to understand the procedure with which the resulting text is composed and this understanding is not dependent on whether the actual execution of the procedure is performed by the author, by the reader or by a computer.

Hand-Picked Combinations

Another oft-cited precursor to generative electronic literature is the book *Cent mille milliards de poèmes* or *A Hundred Thousand Billion Poems* by Raymond Queneau. It is a physical book containing the text of ten sonnets. Each page is cut into parts so that every line of each sonnet appears on its own sliver of paper, allowing the reader to combine lines of different sonnets together to form new poems. The sonnets themselves have been constructed specifically to match each other to form a syntactically and conventionally correct poem with every combination,

> To wit, a reader can construct alternate poems by reading the first line of any of the original sonnets, followed by the second line of any sonnet, followed by the third line of any sonnet—and find that the whole work is artfully constructed so that any reading of this sort produces a sonnet that functions syntactically, metrically, and in its rhyme scheme. And here we see combinatory literature as (independently) discovered by a writer.[18]

Although in a physical format, Queneau's work shows a clear understanding of combinatory text production and the process itself is also easily reproducible by computers.[19] However, what separates the work from computerised literature is the fact that extreme attention has been paid for the source material to match up every time a reader composes a new poem, "Queneau's Poems,

---

16 Strickland, "Writing the Virtual."

17 Burroughs, "The Cut Up Method."

18 Wardrip-Fruin, "Five Elements," 37.

19 As has been done by Nicholas Gessler in a C++ simulation titled "Sonnet." (N. Katherine Hayles, *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005), 115–116. PDF e-book.)

on the other hand, is a high-wire act of writing. He has created a process, but a process that only works when real attention is given to the words."[20] The process has certainly also been informed by experiments in computation and combinatory literature as the book starts with a quote "Only a machine can appreciate a sonnet written by another machine"[21] from Alan Turing, who created a model that led to the development of the modern computer.[22]

Although the functioning of Queneau's procedure relies heavily on the well-crafted material, it does manage to force the reader to pay attention to both the materiality and conventions of language as well as the roles of the author and the reader, "Cent Mille Milliard de Poèmes effectively mocks the theoretical notions of writer and reader, while the power of the text is cleverly demonstrated. (What it does to our notion of the sonnet is perhaps better left unsaid.)"[23] The notion of the sonnet is certainly shattered as "Queneau subverts the rigorous classicism of the sonnet form and its Alexandrine meter through slang colloquialisms and through stereotypes sarcastically perpetuated in the poems. His preface credits the playful form of the book to children's books and disclaims any influence from surrealist games."[24]

Queneau was the founding member of the French literary group Oulipo alongside the mathematician François Le Lionnais.[25] As Florian Cramer notes, his interest in encyclopaedic methodology is already apparent in the works that he produced prior to the *Cent mille milliards de poèmes*, "Earlier in 1947, Queneau had spelled out his own obsessions with mathematics, encyclopedism and street slang in the *Exercices de Style*, a narrative of one short everyday scene in 99 different stylistic variations."[26] This encyclopaedic, repetitive nature with every iteration producing a slightly altered text is also very apparent in specific types of electronic literature, not the least in works that focus on procedural text generation.

The Oulipo also spun off a group interested in algorithmic generation of language called ALAMO (*Atelier de Littérature Assistée par la Mathématique et les Ordinateurs*, workshop of literature assisted by mathematics and computers).[27] Philippe Bootz attributes the ALAMO as the creator of "the automatic generator form" in Europe[28] although it will be pointed out later in this chapter that computer-generated text and automatic textual generators appear in Europe as early as the first computers. "Nowadays, Oulipo focuses on improvisational, non-computational games like the writing of poems in between two subway stops," notes Cramer, as the group ceased to focus on computational methods after the creation of ALAMO and the deaths of Queneau and other active members.[29]

---

[20] Wardrip-Fruin, "Five Elements," 37.

[21] Raymond Queneau, "100,000,000,000,000 Poems," trans. Stanley Chapman, in *Oulipo Compendium*, eds. Harry Mathews and Alastair Brotchie (London: Atlas Press, 1998), reproduced in *The New Media Reader*, eds. Nick Montfort and Noah Wardrip-Fruin (Cambridge, MA: The MIT Press, 2003), 149, accessed March 16, 2017, http://nickm.com/classes/the_word_made_digital/2014_spring/12-oulipo-p.pdf.

[22] Cramer, *Words Made Flesh*, 90.

[23] Aarseth, "Nonlinearity and Literary Theory," 67.

[24] Cramer, *Words Made Flesh*, 89.

[25] Bootz, "The Functional Point of View," 316, footnote. Cramer, *Words Made Flesh*, 89.

[26] Cramer, *Words Made Flesh*, 89.

[27] Bootz, "The Functional Point of View," 308. Cramer, *Words Made Flesh*, 91.

[28] Bootz, "The Functional Point of View," 308.

[29] Cramer, *Words Made Flesh*, 92.

Genius Can Be Mechanised

As the works discussed above show, procedural works do not require digital computing in order to be combinatory and performative. Dada, Burroughs, Gysin and Oulipo all invented procedures in order to examine text and to extend language outside of human creativity through randomness and repetition. These procedures can point out the materiality of language and how literary and artistic forms rely on ossified formalisms that can easily be generated with the right set of instructions, performed by the reader, a performer or a computer. Genius, like it appears in the sonnet form or religious dicta, can certainly be mechanised.

Early Computers Writing

Without delving too much into the history of early computers, which begins in the 19th century with Charles Babbage's *Analytic Engine* and other machines capable of performing different kinds of mathematical calculations and logical operations,[30] it is worth noting when digital computers got their start and how soon they were already being employed for much more than what they were originally envisioned to be used for, that is cracking naval cyphers and running calculations of physics, for example. Alan Turing, one of the founding fathers of computer science with his 1937 conceptualisation of the *Universal Machine* capable of "simulating the activities of any other calculation machine",[31] worked in groups that developed the earliest computers, first the *Colossus* for the cryptography unit of the British government during WWII and later the Manchester computers, which are commonly regarded as the first computers in the world.[32] The first industrially manufactured version of the Manchester computers was the *Ferranti Mark I*, for which "Turing wrote the programming manual [...] and produced a random number generator that produced truly random digits from noise."[33] This random number generator would play a great role in the first work of electronic art and is conceptually important still today as modern procedural text generators also often rely on some sort of a random number generator.

"Yours ardently—M. U. C."

Christopher Strachey, a schoolmaster-turned-programmer, was an acquaintance of Turing's from his days at King's College, Cambridge.[34] Due to the social connection, Strachey was able to procure a copy of Turing's manual for *Mark I* and he quickly made a name for himself in Manchester by adapting the game of checkers for the *Mark I* as well as by programming the computer's speaker to play songs, "The speed and ease with which Strachey appeared to work the Mark I cemented his reputation overnight, and he would soon become known as the man who wrote 'perfect

---

30 Wardrip-Fruin, "Five Elements," 30.

31 Ibid.

32 Ibid., 31.

33 Ibid.

34 Jacob Gaboury, "A Queer History of Computing: Part Three," *Rhizome* (blog), April 9, 2013, accessed October 30, 2016, http://rhizome.org/editorial/2013/apr/09/queer-history-computing-part-three/. Wardrip-Fruin, "Five Elements," 32.

programs,' which would lead to a job offer at the National Research and Development Corporation (NRDC) the following year."[35]

After becoming employed at the NRDC in June 1952, Strachey created his love letter generator[36], which Wardrip-Fruin dates to be "the first piece of digital literature, and of digital art, predating by a decade the earliest examples of digital computer art from recent surveys."[37] Strachey's generator is a program that follows a simple textual algorithm to produce a letter with a salutation, five sentences of varying lengths and an affectionate closing signature from M. U. C. (Manchester University Computer). The sentences follow two main templates with more randomness added by the potential exclusion of adverbs and adjectives, "The first is '*My*—(adj.)—(noun)—(adv.)—(verb) *your*—(adj.)—(noun).' There are lists of appropriate adjectives, nouns, adverbs, and verbs from which the blanks are filled in at random. [...] The second type is simply, '*You are my*—(adj.)—(noun)."[38] This looping algorithm leads to love letters like the ones submitted by Strachey for the arts journal *Encounter*,

> Darling sweetheart
> You are my avid fellow feeling. My affection curiously clings to your passionate wish. My liking yearns for your heart. You are my wistful sympathy: my tender liking.
> Yours beautifully
> M. U. C.[39]

A few different kinds of readings have been given of the love letters. After posting samples of the letters on the notice board of the university's computer science department,[40] Strachey submitted examples and discussion of his procedure to the arts journal *Encounter*, where his essay, suggesting that the work ought to be analysed at the procedural level, rather than based on individual samples of the letters, was printed alongside well-known literary authors.[41] A queer reading, which takes into account that both Strachey and Turing were homosexual, has been suggested due to the lack of the word 'love' in the published letters, calling the program "a love-letter generator that 'could not speak its name'."[42] This reading, however, has been complicated by a later examination of the Strachey archives, where his notes for the algorithms include several variations of the word 'love'.[43] The very formal structure of the letters does not manage to glorify the language used, especially when reading a larger batch of samples. Rather, the repetitive structure and empty adjectives seem to attempt to convince the reader of the vacuousness of expressions that are often used in love confessions, much like Queneau does with the sonnet form. In fact, Turing's biographer notes that the letters were, indeed, amusing to the pair, "Those doing real

---

35 Gaboury, "A Queer History."

36 Modern adaptations of the generator have been created by Matt Shepton and Nick Montfort. Shepton's version can be viewed at http://www.gingerbeardman.com/loveletter/ and Montfort's at http://nickm.com/memslam/love_letters.html, accessed March 28, 2017.

37 Wardrip-Fruin, "Five Elements," 32.

38 Christopher Strachey, "The 'Thinking' Machine," *Encounter* 3, no. 4 (1954): 26–27, quoted in Wardrip-Fruin, "Five Elements," 32–33.

39 Strachey, "The 'Thinking' Machine," 26, quoted in Wardrip-Fruin, "Five Elements," 33.

40 Gaboury, "A Queer History."

41 Wardrip Fruin, "Five Elements," 32.

42 Ibid.

43 Gaboury, "A Queer History." Wardrip-Fruin, "Five Elements," 50, endnote.

men's jobs on the computer, concerned with optics or aerodynamics thought this silly but ... it greatly amused Alan and Christopher."[44]

Wardrip-Fruin follows Strachey's suggested method of reading and goes on to describe how the programmer "understood the other side of combinatory literature—the view of the system itself when one steps back from the individual outputs, the remarkable diversity that can be produced from a simple plan [...] It is a work that can be understood, in fact, as a system—never by an exhaustive reading of its texts."[45] What can already be seen from the templates cited above, but is even more visible in the published examples of poems and the adaptations of the love letter generator for modern day computers, is that even a simple procedure describable in a few lines of natural language (and not too many lines of programming code) is able to yield a very large variety of output texts, of which Wardrip-Fruin notes,

> That is to say, Strachey had discovered and created an example of, the basic principles of combinatory literature—which still lie at the heart of much digital literature today (and, less commonly, non-digital works). Combinatory techniques allow a relatively small number of initial materials to be arranged, following certain rules, into a vast number of possible configurations.[46]

This goes to say that Christopher Strachey could certainly be seen as the first artist working in the digital media, having ported the procedures of combinatory literature to one of the first computers in the world.

Computerised Poems

Although Strachey's experiments were presented in an arts journal and at least cursorily discussed in artistic terms, another mathematician and computer scientist Theo Lutz is widely regarded as the creator of the first computerised poem. Lutz's program *Stochastic Texts*[47] was also presented in a literary journal, Max Bense's journal *Augenblick*, with the author's description of the generative procedure as well as a sample of its output.[48] *Stochastic Texts* utilises the random number generator of the *ZUSE Z 22* computer of the Technische Hochschule Stuttgart to create pairs of logical statements based on a list of "16 subjects and 16 predicates [...], selected from F. Kafka's 'Das Schloss' ('The Castle')."[49] The elementary propositions are connected by one of four different logical constants and further variability is added by adding modifiers to the subject, producing a great variety of sentences like,

---

44 Andrew Hodges, *Alan Turing: The Enigma* (New York: Walker, 2000), 478, quoted in Wardrip-Fruin, "Five Elements," 33–34.

45 Wardrip-Fruin, "Five Elements," 34.

46 Ibid., 33.

47 A modern version in English can be viewed at https://nickm.com/memslam/stochastic_texts.html, accessed April 25, 2017. It has been created by Nick Montfort based on Helen MacCormack's translation of the work.

48 Nick Montfort, "Conceptual Computing and Digital Writing," forthcoming in *Postscript: Writing After Conceptual Art*, ed. Andrea Andersson (Toronto: University of Toronto Press, 2017), accessed April 1, 2017, http://hdl.handle.net/1721.1/92876.

49 Theo Lutz, "Stochastic Texts," trans. Helen MacCormack, Stuttgarter Schule, 2005, accessed April 1, 2017, http://www.stuttgarter-schule.de/lutz_schule_en.htm.

NOT EVERY LOOK IS NEAR. NO VILLAGE IS LATE.
A CASTLE IS FREE AND EVERY FARMER IS FAR.
EVERY STRANGER IS FAR. A DAY IS LATE.
EVERY HOUSE IS DARK. AN EYE IS DEEP.
NOT EVERY CASTLE IS OLD. EVERY DAY IS OLD.
NOT EVERY GUEST IS ANGRY: A CHURCH IS NARROW.[50]

In his article, Lutz makes notes similar to Strachey about how a simple algorithm can create great variability even with very small lists of words. In addition, he also contemplates possible expansions to the algorithm through the inclusion of word frequencies and transition probabilities in the manner of Markov chains to produce sentences that follow the natural use of language. In the end of the article, Lutz envisions that the use of computers will soon achieve "great success in language research and analytical language areas."[51] What is notable in Lutz's article is that it also does not treat textual generation in artistic terms and focuses on the exploration of the new technologies and their potential instead.

Allison Knowles and James Tenney's work *A House of Dust*[52] is another early computerised poem. In 1967, when the work was created, Knowles was associated with the Fluxus group and the work grew out of experimentation in one of the group's seminars, "The work had its beginnings at an informal Fluxus seminar in 1967 in which Tenney, who had been a composer-in-residence at Bell Labs in the early '60s, demonstrated how the Fortran programming language could be employed in chance operations in artmaking."[53] What makes *A House of Dust* different from Strachey and Lutz's experimentation is that the program built by Knowles and Tenney was created, used and presented in relation to an artistic practice.[54] The work is very simplistic and regular in its form: each stanza produced on the dot matrix printer of the *Siemens 4004* describes a house built of ordinary and extraordinary materials in varying locations,

A HOUSE OF STRAW
IN AN OVERPOPULATED AREA
USING ELECTRICITY
INHABITED BY PEOPLE WHO SLEEP ALMOST ALL THE TIME[55]

Knowles' poem is actually much simpler than works by Strachey and Lutz: containing no randomisation of forms, all the variability of the output relies on the carefully crafted lists of words, which carry surprising human-authored pairs like "inhabited by French and German speaking people" and "inhabited by friends and enemies."[56] Knowles herself did go on to execute a

---

50 Ibid.

51 Ibid.

52 Modern adaptations of the work have been created by Nick Montfort and Zach Whalen. Montfort's version can be found at http://nickm.com/memslam/a_house_of_dust.html and Whalen's at http://zachwhalen.net/pg/dust/, accessed March 28, 2017.

53 Scott Taylor, "Alison Knowles, James Tenney and the *House of Dust* at CalArts," *24700: News from California Institute of Arts* (blog), September 10, 2009, accessed November 8, 2016, http://blog.calarts.edu/2009/09/10/alison-knowles-james-tenney-and-the-house-of-dust-at-calarts/.

54 Montfort, "Conceptual Computing and Digital Writing."

55 Allison Knowles and James Tenney, *A House of Dust*, 1967, accessed November 8, 2016, http://beinecke.library.yale.edu/about/blogs/room-26-cabinet-curiosities/2010/02/15/house-dust.

56 Ibid.

version of the house in New York City, turning one of the stanzas created by the machine into a real-life public sculpture, sound work and community space while working as a Guggenheim fellow in 1968.[57]

"The First Book Ever Written by a Computer"

The above title is a quote from the cover of *A Policeman's Beard Is Half-Constructed*, which claims that the 1984 book has been written by an interactive dialogue-generator program named *Racter*. *Racter* itself is the work of programmers William Chamberlain and Thomas Etter, intended for commercial use at home and significantly expanded for the production of texts in the *Policeman's Beard*.[58] In his introduction to the book, Chamberlain claims, "[T]he writing in this was all done by a computer. The book has been proofread for spelling but other is completely unedited,"[59] and later continues, "In this way, certain aspects of English are entered into the computer. This being the case, the programmer is removed to a very great extent from the specific form of the system's output. The output is no longer preprogrammed form. Rather, the computer forms the output on its own."[60]

Composed of small fragments in a variety of forms ranging from dialogues between characters (or transcriptions of dialogues between the user and the *Racter* program) to short poems and longer prose that never exceeds a single page, the book presents themes like love, nature, electricity as well as both human and machine consciousness. Enhanced by the occasional self-reference pointing out the electronic source of the texts in statements like "More than iron, more than gold I need electricity,"[61] the machine-like nature is also visible from the fragments which utilise all kinds of logical connections between concepts in their narration. Take, for example, the fragment that appears right at the beginning of the book,

> We will commence with a question: does steak love lettuce? This question is implacably hard and inevitably difficult to answer. Here is a question: does an electron love a proton, or does it love a neuron? Here is a question: does a man love a woman or, to be specific and to be precise, does Bill love Diane? The interesting and critical response to this question is: no! He is obsessed and infatuated with her. He is loony and crazy about her. That is not the love of steak and lettuce, of electron and proton and neutron. This dissertation will show that the love of a man and a woman is not the love of steak and lettuce. Love is interesting to me and fascinating to you but it is painful to Bill and Diane. That is love![62]

From the mechanical connections of variables with the formats like 'X loves X' like 'Y loves Y' and the amplification of adjectives from loving to infatuation and obsession, it is obvious that computation has clearly played a part in the creation of the text. However, how great a part computation plays in the creation of these particular texts is another question. The analogies

---

57 Taylor, "Alison Knowles, James Tenney and the *House of Dust* at CalArts."

58 Aarseth, *Cybertext*, 12, 129.

59 William Chamberlain, *The Policeman's Beard Is Half-Constructed: Computer Poetry and Prose by Racter* (New York: Warner Books, 1984), n.p., August 10, 2014, accessed November 2, 2016, https://archive.org/details/pdfy-T3abGAQ80ecd63PL.

60 Ibid.

61 Ibid.

62 Ibid.

between love, subatomic particles and steaks and lettuce have surely required quite some curation from Chamberlain and it is reasonable to question to what extent these unique connections have been put together by a computer.[63]

Espen J. Aarseth questions the origins of the book,

> It is therefore reasonable to assume that, contrary to Chamberlain's claim in the introduction of the book, Racter did not 'write' it without a substantial amount of help, a suspicion that is confirmed by John Barger's examination of the boilerplate (fill-in-the-blanks) system that allows Racter to form well-constructed sentences. As Barger (1993) points out, the 'wacky' style of Racter's output is really Chamberlain's own, the product of a clever human writer posing as a clever program.[64]

Aarseth calls this type of collaborative authorship shared between a computer program and a human 'cyborg' literature and notes that "it can be safely assumed that the architect, selector and editor of the texts is human."[65] In the case of the *Policeman's Beard*, the human-computer cooperation appears in two stages of the process as the human author is involved in both preprocessing (writing of the code) and postprocessing (editing) of the book's contents.[66] What Aarseth's analysis and the concept of cyborg literature show is that the boundaries of authorship are muddy at best when it comes to computerised literature. We must pay attention to the syntax devised by programmers, the grammars collected by artists and the choices made in copy editing as much with the earliest works like the love letters and the *Policeman's Beard* as with the bots and internet art of today. Aarseth's note in the beginning of his analysis of the *Policeman's Beard* still applies, "With so many unspecified variables behind the general idea of computer as author, the question, Who or what writes? can only be decided case by case."[67]

Understanding the Procedure

As noted above, combinatory texts have been used in religious rituals and literary experimentation hundreds of years prior to the advent of mechanical universal calculators and digital computers. However, it is also worth noticing that as soon as digital computers were invented, they have been used for literary experimentation in addition to ciphering texts and performing massive calculations. What is common in both the physical literary experiments of the Dadaist, Beats and the Oulipo and the poetry generated with early computers is that they all highlight the process of generating the text using a combinatory algorithm. What can be discovered from the processes of Queneau, Strachey and Lutz is one of the central characteristics of combinatory literature: a very large amount of uniquely diverse outputs can be generated from a fairly small selection of words with a well-designed procedure. Thus, these kinds of combinatory works can never be fully understood by simply reading their output. Rather, the full understanding of the works requires

---

[63] Nowadays, there are services like Wordnet and ConceptNet which include machine-usable knowledge about the connections between words and concepts, which can easily be utilised to produce texts like Chamberlain's.

[64] Aarseth, *Cybertext*, 132.

[65] Ibid., 134.

[66] Ibid., 135.

[67] Ibid., 132.

an understanding of the work as a system, as a performative process that generates the works, which applies also to the other earlier examples, including the making of the Dadaist poem. However, sometimes the output can also deceive the analysis, as does the presentation of Chamberlain's *Policeman's Beard* as the first novel written by the computer.

Furthermore, what can be seen from the comparison between the more physical kinds of combinatory procedures and their digital counterparts is that understanding the processes of computerised poetry does not require understanding how the computers were programmed. Rather, the procedures designed by Strachey, Lutz, Knowles and Tenney are so simple that they could easily be performed without computers, by shuffling cards with the words written on them or by the roll of the dice, although computers make the generation of tens and hundreds of unique output texts much faster. This shows that understanding combinatory text generation is not really about understanding the code but more about understanding the malleability of language, the shifting meanings assigned to individual words and the formulaic nature of the everyday language that surrounds us. These same aspects are still at the heart of understanding the more contemporary examples of procedural works. What is important is the understanding of the procedure, regardless of whether it is performed by the author, the reader or the computer.

3. Reading Electronic Literature

Authoring texts with intermediary grammars that a software program compiles into randomly assembled outputs raises questions about the authorship of those resulting pieces of writing. It might be the first instinct to attribute some creativity of the writing process to the computer software although all the rules and data that the program acts upon are designed by a creative human. For Philippe Bootz, who in his article "The Functional Point of View: New Artistic Forms for Programmed Literary Works" develops a model to describe the different factors in the meaning-making process of generated text, the answer to these questions of authorship is very clear,

> Should the computer be considered an artificial ghost-writer? Such an idea might more quickly come to minds when dealing with generators, for in this case the reader, having been invited to take a part in it, clearly identifies that the writing is not completed by the author. For me, the answer to this question is a definite 'No.' The induced data [by the reader] have all the characteristics of data—that is, only their content is generated; that is not the case for either the definition of their structure or for orders that might be added to the corpus of the source and would then expand its possibilities.[1]

Florian Cramer, in his survey of generative art both seconds Bootz's attribution of creative work to the author and manages to explain the reason for the appearance of these questions in criticism of generative artworks,

> When [the digital artist] Cornelia Sollfrank states that 'a clever artist makes the machine do the work,' it still implies that the artist *makes* it work in the first place. Poiesis, making, becomes a second-order poiesis of making something that makes something else. So poetry, making, turns into poetics, the making of making. When making turns into meta-making, subjectivity simply shifts to a second order position, residing in the formula instead of the product. This fact is repeatedly ignored by critical observers whose perspective remains fixated on the product and who wrongly conclude, in a fallacy reminiscent of Plato's cave, that technology has done away with the subject behind the work.[2]

By applying traditional literary theory to works of electronic literature, it is easy to end up analysing the product and its immediate source instead of looking at the authoring process as a whole. Many scholars emphasise that reading electronic literature should take a more thorough approach to the process of authoring the works rather than applying traditional genres and models to their analysis. Aarseth notes that by failing to do so "investigation into these new ergodic forms

---

1 Bootz, "The Functional Point of View," 315.

2 Cramer, *Words Made Flesh*, 87. Sollfrank quoted from Cornelia Sollfrank, ed. *net.art generator* (Nürnberg: Verlag für moderne Kunst, 2004), 87, 134.

that will emphasise how they differ from narrative media"[3] will be hindered. Noah Wardrip-Fruin calls for an analysis that does not merely consider electronic literature through the digital medium but rather looks at the processes that bring about the texts, "I think it is important to distinguish process and surface, rather than collapse both into the 'medium,' in part because I believe that a major next step for our field is to begin to interpret processes."[4]

Many of the scholars mentioned above offer theories about the processes that come into play when generation is assisted by computers. In particular, this chapter will focus on finding frameworks that can be utilised when reading bots as examples of electronic literature. Bootz's functional point of view will be discussed most thoroughly as it most directly deals with the questions related to text generators. Furthermore, more widely discussed theories of Aarseth and Wardrip-Fruin will be taken into account by comparing how well their models can aid the analysis bots in particular.

The Functional Point of View

As noted in the introduction, Philippe Bootz's functional point of view defines text "mainly as a process and not only as the object the reader sees on screen,"[5] which is clarified later in the article, "This fact incites us to think that literature is not only the 'result' of the program, i.e. the text (in a classical meaning) that appears on the screen (named 'text-to-be-seen' in this paper), but also, and more importantly, the process that goes on by itself to produce this text-to-be-seen."[6] The functional point of view describes "the complete communication chain (called 'work' later on) between an author, defined as the initiating subject of the communication process, and a reader, defined as the targeted subject of this process."[7] What is essential here is that the work consists of the whole communication process all the way from the author's idea of the work to the reader's understanding of it.[8]

Bootz outlines several intermediary types of texts that appear in this communication process. The term 'written-text' denotes the author's idea of the work, "the author's project before any description is given of it."[9] While the written-text "is the most abstract textual object, expressed only through the multiple descriptions that the author may make of it: discursive, symbolic, graphic or other, which may or may not be designed for a computer,"[10] the writing process performed by the author translates the idea of the work into a set of rules and functions shared between the author and the text generator, which moves the text into the generator's domain. The resulting text is the 'author-text,' which "describes the elements of the written text that are necessary for the generation of terms that are understandable both by the author and by the agent of the generation function."[11] The author-text is composed of two types of objects, namely the source and the data. Bootz uses the term source with nearly the same meaning as source code,

---

3 Aarseth, *Cybertext*, 141.
4 Wardrip-Fruin, "Five Elements," 48.
5 Bootz, "Functional Point of View," 307.
6 Ibid., 308.
7 Ibid., 309.
8 Ibid., 310.
9 Ibid.
10 Ibid.
11 Ibid.

"an ordered structure of orders."[12] Data, on the other hand, is described as "a series of materials whose structures are set forms but are not the contents."[13]

This division between the source and the data is similar to Wardrip-Fruin's distinction between process and data, which he has adopted from game theorist Chris Crawford's idea about process intensity in different kinds of computer software. Wardrip-Fruin classifies "words, images, and sounds" as data, which are processed through "algorithms and calculations."[14] Furthermore, different programs make use of data and processes to differing extents, which is something that should be considered when analysing different works of electronic literature. Crawford clarifies these differences between programs by separating them into process-intensive and data-intensive programs, "A process-intensive program spends a lot of time crunching numbers; a data-intensive program spends a lot of time moving bytes around."[15]

While many bots and text generators are fairly simple programs, their process intensity can be assessed by the amount of boilerplate text used to create the resulting texts. Firstly, if the bot or text generator makes use of an external data source, weather or stock data, for example, the processing of this data plays an important role in the creation of human-readable text. Secondly, when using a grammar- or tile-based approach to textual generation, the process intensity can be evaluated by discerning the extent to which readymade boilerplate formulas are used. From the examples of early electronic literature presented in the previous chapter, *A Policeman's Beard Is Half-Constructed* could be said to be less process-intensive due to its extensive human-sourced boilerplate text than Strachey's love letter generator the logic of which can be represented with a few lines of text and simple lists of individual words. Clearly, neither of these examples requires intensive computation as they work with very short texts; the differences between process- and data-intensity are nearly negligible in these cases. However, by looking at the logic and design of these programs, it is possible to discern differences in the ratio of use between process/source and data.

The next form of text within a work of electronic literature is created while running the generator. When the program is run, the author-text, coded as data and source, is turned into a 'text-to-be-seen,' the output text that is presented to the reader. The text-to-be-seen is "spatiotemporal [...] and attached to the medium of the screen," meaning that it is compiled on demand and accessible only while the program is running.[16] This temporary existence is complicated by the nature of bots that post their creations on social media, which maintains an archive of produced texts even when the program is not running. Bootz also complicates this by the introduction of something called 'sentence-texts,'

> One can spot, in the text-to-be-seen, whatever it may be, *sentence-texts*, defined as text objects that the reader would obtain by copying the whole of the sentences that are offered in the text-to-be-seen. In no case whatsoever can the characteristics of a 'text on computer' be reduced to

---

12 Ibid.

13 Ibid.

14 Wardrip-Fruin, "Five Elements," 38.

15 Chris Crawford, "Process Intensity," *Journal of Computer Game Design* 1, no. 5 (1987), quoted in Wardrip-Fruin, "Five Elements," 37.

16 Bootz, "The Functional Point of View," 310.

> those of the sentence-texts, as far as the work's meaning or style or literary nature is concerned.[17]

When analysing bots according to Bootz's model, these problems might be negotiated by approaching the texts published on social media as the text-to-be-seen. After all, the publication medium for the bots is, indeed, the electronic screen but, more specifically, the work is designed and planned to be presented and read within the frame of social media networks and the source of the author-text reflects this. This fact would suggest that the only true way to encounter the text-to-be-seen of textual generator bots is on social media. An analogy to the copied sentence-texts, in this case, would be the presentation of tweets or other social media posts outside of their original social media context as quotes in academic works or news articles, for example.

The text-to-be-seen bridges the communication process from the generator's domain to the reader's domain. However, the text-to-be-seen is not the same as the text read and understood by the reader and, thus, the final stage of the communication process is called the 'read-text,' which is "the mental representation of the work (as a result of the complete communication process, and not merely the text-to-be-seen) that the reader brings into being."[18] The read-text results from everything that is read, understood (with the aid of the author's description of the work) and inferred from the work and its processes.

There are several factors that make the read-text different from the text-to-be-seen and the author-text. In the case of generative art, the generator can be programmed to create texts that are randomised, "computationally variable," based either on randomness internal to the computer or on external sources, or the texts can be determined, "computationally fixed" in a way that produces the same text each time the generator is run.[19] With computationally variable generators, the reader faces a different text at each reading and different readers are likely to encounter different texts-to-be-seen. Furthermore, if the program is designed to run only once, without the possibility of either returning to the generation process later on or with the possibility of returning only to the text generated at the first run, "the reader has no way of discerning the difference between the constant elements and those that are calculated inside what he or she is reading."[20] This ability to discern the constant templates from the changing elements is of extreme importance in analysing bots, as will be shown in chapter 5.

The theory that Aarseth introduces in his book *Cybertext* describes the generative process based on how the texts authored by the writer can be accessed by the reader. Aarseth calls the different access mechanisms traversal functions. These traversal functions compile snippets of text created by the author, textons in Aarseth's terminology, into texts accessible to the reader, scriptons.[21] These terms have parallels in Bootz's theory: textons replace Bootz's data, traversal functions the source and the compiled scriptons the text-to-be-seen. A cybertext can be either static or dynamic, computationally fixed or varied. A dynamic text can either be intratextonically dynamic, "the contents of scriptons may change while the number of textons remains fixed," or textonically dynamic, "the number of textons may vary as well."[22] A text generator based on randomness is most commonly intratextonically dynamic, which introduces instability in the

---

17 Ibid.

18 Ibid.

19 Wardrip-Fruin, "Five Elements," 41.

20 Bootz, "The Functional Point of View," 312.

21 Aarseth, "Nonlinearity and Literary Theory," 60.

22 Ibid., 61.

reading process. The reader is rarely as active a participant in textual creation with generators as they are with interactive or hypertext fiction, the types of e-lit Aarseth mainly focuses on, but the following still applies to the reading process of generative art as well,

> [T]he understanding (beyond trivial) of a nonlinear text can never be a consummate understanding, because the realization of its script (and not just its meaning) belongs to the individual user, who is acutely aware of his or her own constructive participation. Since the object is unstable in a syntactic and semantic sense, it cannot be read, only glimpsed and guessed at.[23]

As the texts-to-be-seen of text generators are unstable and potentially changing with each reading, the reader never has complete access to the author-text, which can merely be "glimpsed and guessed at."

This guessing added to another, more traditional source of uncertainty in analysing even the most linear literary works, the uncertainty of meaning-making based on the read text, adds more variety in the representation of the work in the reader's mind. The reader might not read all of the text available for their reading. This also occurs with traditional narrative books with the reader skipping passages.[24] With interactive or hypertext fiction, the reader will likely explore enough passages to gain an understanding of the central narrative in the work.[25] When faced with a large mass of (generated) text, the reader is also likely to glimpse around without reading every single word, just like they would do with an internet news article or a blog post.

Aspects to Analyse

What Bootz's model manages to show about generative works is that the major creative act lies at the engineering of the author-text, the selection of the data and the design of the processes that create the desired text-to-be-seen. Taking into consideration the sources of uncertainty that affect the reading of the works, as described above, how should different works of electronic literature be read, analysed and critiqued? Firstly, it is important to note that the readers of the text do not need to read the same words to consider the texts-to-be-seen to be representative of the same work,

> The important lesson to be learned from discontinuous and forking texts is that when two readers approach a text they do not have to encounter the same words and sentences in order to agree that it probably was the same text. And this is not new: it is classical feature of reading, as Roland Barthes points out in his comment on tmesis (1975).[26]

This points out, as does Bootz's modelling of the creative process, that instead of looking only at the texts directly encountered by the reader, the critique of electronic literature has to take into

---

[23] Ibid., 65.

[24] Aarseth, *Cybertext*, 47, 74.

[25] J. Yellowlees Douglas, "'How Do I Stop This Thing?': Closure and Indeterminacy in Interactive Narratives," in *Hyper / Text / Theory*, edited by George P. Landow (Baltimore: Johns Hopkins University Press, 1994), quoted in Hayles, *My Mother Was a Computer*, 165.

[26] Aarseth, *Cybertext*, 74.

consideration also the processes that create these texts-to-be-seen. The following sections will elucidate the analysis of the different aspects of the author-text by looking at the works of computational artists Nick Montfort, Stephanie Strickland and Mark Sample. These works have been chosen as examples of modern works of electronic literature because they utilise and approach text in a variety of ways, are published on the web for anyone to read and contain clearly annotated source code for analysis.

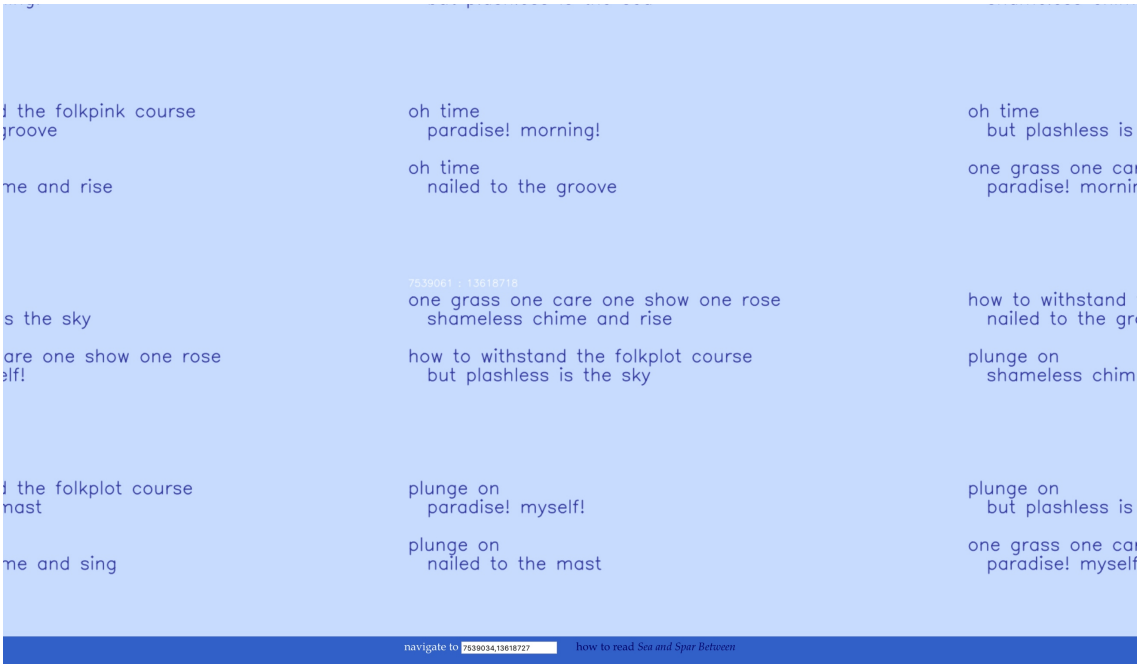

Figure 3. A screenshot of Nick Montfort and Stephanie Strickland's Sea and Spar Between. *(Accessed April 22, 2017.)*

Figure 4. A screenshot of Nick Montfort's Flow My Tears. *(Accessed April 22, 2017.)*

Reading the Source

Many of the seminal publications in the field of electronic literature in the 1990s and early 2000s suggest viewing the source as one of the methods of evaluating the processes that account for the generated text. With works published as web pages, the reader is able to view the source of the web page in addition to the actual page that appears in the browser window, "An experienced reader of HTML, if reading code along with the browser version of a document, will concurrently be able to read the commented version, effectively gaining access to more than one version of the same text."[27] This kind of viewing of the source is not only limited to works based on web technologies but can be useful when evaluating command line based generators whose source codes have been shared by artists to be run by the readers on their own computers. There are several aspects to consider when reading the source code: Firstly, it might contain helpful comments to guide the understanding of the work.[28] However, it should be noted that there are also artistic choices that the authors make with the source code and, thus, it ought to be considered as an integral part of the work instead of a merely practically motivated method of giving orders to a computer. For example, it should be noted if the comment blocks guide the reader to pay attention to certain parts of the work and if they introduce some themes or concepts that would otherwise be unnoticed when viewing the work. All in all, the analysis should note how the source code guides the reader's experience of the work and what the code adds to the viewing of the work in a browser.

Sea and Spar Between[29] is a web-based poetry generator created by Nick Montfort and Stephanie Strickland with an attitude of digital humanities research. It presents the viewer with a large grid of four-line poems with two stanzas which combine the most commonly used words and tropes of Emily Dickinson with those of Herman Melville. This grid extends outside of the viewer's screen and the viewer can navigate between the poems in the grid using the keyboard or the mouse, although any systematic examination of the grid is made impossible both by the huge amount of poems and by the work's hypersensitivity to mouse movements, since even a slight movement of the mouse is enough to jump several hundred poems ahead in the grid. What makes *Sea and Spar Between* particularly interesting is the annotated source code from which it is possible for the reader to deduce patterns that appear in the systematically generated 'lattice' of poems on the website. Reading the source, the audience learns that the size of the grid has not been chosen randomly. Rather, the work "defines a space of language populated by a number of stanzas comparable to the number of fish in the sea, around 225 trillion."[30] There are other notes about the nature of patterns included in the generator like "Dickinson's poems include many words

---

[27] Loss Pequeño Glazier, *Digital Poetics: The Making of E-Poetries* (Tuscaloosa: The University of Alabama Press, 2002), 107.

[28] Hayles, *My Mother Was a Computer*, 54.

[29] Nick Montford and Stephanie Strickland, *Sea and Spar Between*, 2010, accessed February 7, 2017, http://www.saic.edu/webspaces/portal/degrees_resources/departments/writing/DNSP11_SeaandSparBetween/index.html.

[30] Montfort and Strickland, *Sea and Spar Between*, source code (sea_spar.js), lines 4–6.

ending in 'less,' such as 'artless.' Some of their stems (such as 'art') follow, grouped by number of syllables."[31]

The comments in the source code of *Sea and Spar Between* focus mainly on the word choices and how they reflect the style of the original authors from whose work they were selected. The commentary leaves the processes of combining those words into lines and stanzas unexplained, commenting only, "Functions: These generate each type of line, assemble stanzas, draw the lattice of stanzas in the browser window, and handle input and other events."[32] This is followed by mere notations for the functions that provide the first and second lines separately, with no notes on the variables or design of the algorithms. A reader might notice the different formats from lines like

```
return 'one ' + dickinsonNoun[0][a] + ' one ' + dickinsonNoun[0][b] +
' one ' + dickinsonNoun[0][c] + ' one ' + dickinsonNoun[0][d];[33]
```

This, when examined in conjunction with the view in the browser, shows that one of the starting lines designed by Montfort and Strickland returns a line punctuated by 'ones' à la Dickinson. However, how the algorithm chooses the individual words based on the coordinates of the poem in the lattice remains shrouded in mystery to the casual reader, due to the unexplained variables and attributes in lines like,

```
function oneNounLine(n)
  {
      var a, b, c, d = n % dickinsonNoun[0].length;
      n = Math.floor(n / dickinsonNoun[0].length);
      c = n % dickinsonNoun[0].length;
      […][34]
```

A remix of *Sea and Spar Between* by Mark Sample, titled *House of Leaves of Grass*[35], utilises the same source code and algorithms but provides a more thorough explanation of both the process and the functions composing the lines. Sample's work uses most frequent and thematically significant words selected from the works of Walt Whitman and Mark Z. Danielewski using computer-based textual analysis, which is a key tool in digital humanities research. Sample's source code explicitly tells the readers about the role of textual analysis in the production of the work, whereas *Sea and Spar Between* does not explain how the words have been chosen, "The words and phrases from both works were selected according to either frequency or thematic significance. Frequency was determined by using standard text-mining tools, in this case Voyant as well as NGramTool."[36] Furthermore, Sample gives examples to the reader of the type of lines the different functions produce in the comments, "The function riseAndGoLine can generate, e.g., 'graspless dance and go'."[37] His comments also make explicit the deterministic nature of both his and Montfort and Strickland's works, "The lattice of House of Leaves of Grass is deterministic; each point of it maps to a particular combination of words and lines so that (in theory) the system

---

31 Ibid., lines 82–83.

32 Ibid., lines 109–110.

33 Ibid., line 128.

34 Ibid., lines 119–123.

35 Mark Sample, *House of Leaves of Grass*, 2013, accessed February 7, 2017, http://fugitivetexts.net/houseleavesgrass/index.html.

36 Sample, *House of Leaves of Grass*, source code (house_leaves_grass.js), lines 89–92.

37 Ibid., lines 231.

can enumerate all possible texts."[38] Sample, like Montfort, is an educator, a professor at Davidson College in North Carolina. His annotations to the source code take an exceptionally educational tone as he also explains the functioning of one of the most basic mathematical operators used in almost any programming language,

> % is the mod operator. "n % m" yields the remainder from when n is divided by m. So, for instance, the value of "n % 10" is at least 0 and at most 9. In this case, % allows any value n to be used to pick an element of the array shortPhrase.[39]

Although Sample's annotations to the code are more extensive and open up the creative process more to the readers unable to understand the actual JavaScript functions, both of these works are rare in that they include extensive annotations in the source code. However, a reader with no knowledge of programming languages can learn from unannotated works as well. The plain text, human-readable language included in the source code of, for example, animated and generative poetry works often contains information that can be helpful in looking at the work in the browser window. *Flow My Tears*[40] is another web work by Nick Montfort, an animated poem which starts by a cascade of the words "flow my tears" in a vertical strip down the screen. At the same time, the lines keep flickering with what seems to be alternative spellings of the phrase, changing their state so quickly that the reader has to pay extra attention to catch the differences between the flickering phrases and the "flow my tears" lines filling the screen in a narrow column. After close examination, the reader is able to distinguish several alternative types of phrases which appear and disappear on random lines on the screen. These phrases include other bodily fluids, "flow my blood" and "flow my phlem," but also phrases similar in pronunciation, "flaw my tears." There are also a couple of longer lines, which extend outside of the uniform-length column of text, like "flow my tears, fall."

The alternate phrases are updated on the page at 150-millisecond intervals: while reading the individual phrases can be difficult for the reader, deducing the exact amount of possible alternatives is almost impossible at this rate. This is where the source code can offer another possibility for a deeper examination of the work even to the readers with no understanding of JavaScript. Although the code does not contain any other annotations than the initial copyright notice releasing the work to be redistributed and remixed for any purpose, as soon as the reader scrolls down, they come across a comma-separated, spaced-out list with all the 12 possible alternatives to the title phrase. The browser's developer tools often highlight the 'string'-type text with its special colour, making it even easier for a non-code-savvy reader to see the parts that concern their experience as a reader the most. Naturally, a reader with more intimate understanding of programming might notice how the author plays with the probability for each alternate's appearance by filling the list with 90 lines of 'flow my tears' with the function,

```
for (i=0; i<90; i++) {
      texts.push('flow my tears');
}
```[41]

---

[38] Ibid., lines 299–301.

[39] Ibid., lines 173–176.

[40] Nick Montfort, *Flow My Tears*, 2015, accessed February 7, 2017, http://nickm.com/poems/flow_my_tears.html.

[41] Montfort, *Flow My Tears*, source code (flow_my_tears.html), lines 92–94.

Furthermore, the code-savvy reader might even change the rate of refresh for the alternate phrases by altering the `setInterval(litany, 150);`[42] command with their browser's developer tools. Although the rate of refresh is important to the experience and probably inaccessible to a reader not accustomed to reading JavaScript source code, reading the highlighted strings in the code is understandable to almost any web user. As the reader learns the possible phrases from the highlighted source code, they are able to concern themselves more with the other aspects of the work: the non-coder can start paying attention to the rate at which the different phrases flicker on the screen, how it affects the atmosphere of the work as well as how it is connected to the theme of the work.

There is an emerging field called critical code studies related to digital humanities that urges a very close and informed reading of source code of all kinds of programs. Its detailed understanding of the aspects that combine in any given program can also be used to guide the analysis of the source code of e-lit works,

> In addition to symbols and characters in the program files themselves, paratextual features will also be important for informed readers. The history of the program, the author, the programming language, the genre, the funding source for the research and development (be it military, industrial, entertainment, or other), all shape meaning, although any one reading might emphasize just a few of these aspects. The goal need not be code analysis for code's sake, but analyzing code to better understand programs and the networks of other programs and humans they interact with, organize, represent, manipulate, transform, and otherwise engage.[43]

The ability to view code also allows for code to be copied and remixed to create new works. This kind of practices are already noted in Loss Pequeño Glazier's 2002 study of works of electronic literature, where he likens them to tropes in poetry, "The ability for 'writing' to circulate in this manner [being copied as code snippets from one work to another] also marks its relation to poetry, as if the JavaScript fragments were tropes or passes in an oral poetry corpus, freely incorporated from one recitation to the next (and just like in oral poetry, subject to permutation each time it is passed on)."[44] Today, rather than sharing snippets of JavaScript to create individual elements like drop down menus and random number generators on web pages, the sharing of source code is related to remixable works whose source code is shared with different kinds of open source licences. These licences allow the remixing of the software in such a way that the underlying logic and presentational aspects of the original work remain the same while the remixing artist creates their own themes by placing different data into the program code.

A great example of this kind of online poetry trope being remixed and re-remixed by different artists is Nick Montfort's project *Taroko Gorge*[45]. Along with the original work, Montfort has published over twenty iterations of the same source code on his website with different themes introduced by other artists. The work was originally an attempt to create a poetry generator about the beauty of nature, "Taroko Gorge originated as a Python program that I developed at Taroko Gorge National Park in Taiwan. If others could go to a place of natural beauty and write

---

42 Ibid., lines 97.

43 Mark C. Marino, "Critical Code Studies," *Electronic Book Review* (December 4, 2006), accessed February 29, 2016, http://electronicbookreview.com/thread/electropoetics/codology.

44 Glazier, *Digital Poetics*, 104.

45 Nick Montfort, *Taroko Gorge*, 2009, in *Electronic Literature Collection* 3 (February, 2016), accessed February 6, 2017, http://collection.eliterature.org/3/work.html?work=taroko-gorge.

a poem about that place, why couldn't I write a poetry generator, instead?"[46] However, like many of his other pieces, Montfort published the work with a very permissive licence, allowing modification and redistribution of the work for any purpose, commercial or non-commercial.[47] This leniency has invited many authors of electronic literature to utilise Montfort's concise source code as a platform to create their own generative poems of the same format, turning Montfort's generator "into an ever-expanding platform for poetic play"[48] and "an authoring system open for many types of generative manipulation."[49]

Montfort's original version creates stanzas of two to four lines expressing the narrator's progress along a path in the Gorge and the sites along it,

> Mists trail the shape.
> Flows dream.
> Forests hum.
> Height paces the ripplings.[50]

These stanzas are separated by "a 'cave' line that trails off, as if into darkness, like the tunnels in the park that were carved by Chiang Kai-shek's Nationalist army."[51] These cave lines contain long lists of adjectives that break off before the object they describe is revealed, "shade the sinuous objective arched cool —"[52]

While Montfort's original piece focuses on nature imagery, creating a unique trip into the Gorge for each of its readers from sparse sets of nouns and transitive and intransitive verbs, its remixes are often everything but calm and restrictive. There are works about metropolises, like Scott Rettberg's *Tokyo Garage*[53], that aim for excess instead of the calm imagery of the original[54] with stanzas like,

> Prostitute eyes the banker.
> Costumed mascots succeed.
> Rat tests the motorcycle gang.
>
> explore the shabby unyielding opaque—[55]

---

46 Nick Montfort, "Taroko Gorge" (Author's Statement), *Electronic Literature Collection* 3 (February, 2016), Accessed February 6, 2017. http://collection.eliterature.org/3/work.html?work=taroko-gorge.

47 Montfort, *Taroko Gorge*, source code (taroko-gorge.html), lines 15–17.

48 "Taroko Gorge Remixes," *Electronic Literature Collection* 3 (February, 2016), accessed February 6, 2017, http://collection.eliterature.org/3/collection-taroko.html.

49 "Scholars Contemplate the Irish Beer" (Editorial Statement), *Electronic Literature Collection* 3 (February, 2016), accessed February 6, 2017, http://collection.eliterature.org/3/work.html?work=scholars-contemplate-the-irish-beer.

50 Montfort, *Taroko Gorge*.

51 Montfort, "Taroko Gorge" (Author's Statement).

52 Montfort, *Taroko Gorge*.

53 Scott Rettberg, *Tokyo Garage*, 2009, in *Electronic Literature Collection* 3 (February, 2016), accessed February 7, 2017, http://collection.eliterature.org/3/work.html?work=tokyo-garage.

54 Scott Rettberg, "Tokyo Garage" (Author's Statement), *Electronic Literature Collection* 3 (February, 2016), accessed February 7, 2017, http://collection.eliterature.org/3/work.html?work=tokyo-garage.

55 Rettberg, *Tokyo Garage*.

There are tens of other works that use the *Taroko Gorge* code more or less in its original form: Kathi Inman Berens uses it "to mash the space of computation with the female, domestic, and tactile,"[56] discussing the place of women in programming scene with references to food in her *Tournedo Gorge*[57]. Flourish Klink, in her remix *Fred and George*[58], utilises "the fairly repetitive and easily distilled elements"[59] of Harry Potter twincest fan fiction sub-genre, the romantic/sexual pairing of the Weasley twins, to illustrate "the internet's endless and endlessly kinky desires" in "an act of trolling."[60]

Although not directed by Montfort in the copyright stipulations of his code, the remixes of *Taroko Gorge* have evolved to respect a certain form. Most of the works pay special attention to follow the format of the original, presenting a similar dualistic structure of multi-line stanzas separated by an individual line, although digressions from the original AA/ABA/ABBA–C–AA/ABA/ABBA pattern[61] are possible. From the first original remix to today's adaptations, the remix most often includes a list of other contributors that precede the author of the remix in their task. This practice was set by Scott Rettberg in *Tokyo Garage*.[62] Furthermore, most of the remix authors have chosen to title their work so that their title closely resembles *Taroko Gorge* in pronunciation and look. This is possible to see from the titles included in the *Electronic Literature Collection*'s selection of remixes: only *Camel Tail*, *Scholars Contemplate the Irish Beer* and *Snowball* jump out between names like *Tasty Gougère*, *Takei, George*, *Toy Garbage* and *Take Ogre*.[63]

Taroko Gorge and its remixes function like modern-day oral poetry: the remixes take the original form, incorporate and modify it to work with a new topic and release it back to the community. Although some aspects, like the notation of the contributors preceding the author in the credits on the right side of the screen, are specific to this one work, the release of code into the public domain invites any technology to be appropriated by other artists and this history might not only be visible on the surface level but also apparent in the source code showing the copyright notices and modifications made by all the other artists preceding the remixer. Therefore, reading source code can also reveal the reader of electronic literature a lot about a work's history and suggest relations to works that the reader might have otherwise ignored in their viewing of the work.

---

56 Kathi Inman Berens, "Tournedo Gorge" (Author's Statement), *Electronic Literature Collection* 3 (February, 2016), accessed February 7, 2017, http://collection.eliterature.org/3/work.html?work=tournedo-gorge.

57 Kathi Inman Berens, *Tournedo Gorge*, 2012, in *Electronic Literature Collection* 3 (February, 2016), accessed February 6, 2017, http://collection.eliterature.org/3/work.html?work=tournedo-gorge.

58 Flourish Klink, *Fred and George*, 2012, in *Electronic Literature Collection* 3 (February, 2016), accessed February 6, 2017, http://collection.eliterature.org/3/work.html?work=fred-and-george.

59 "Fred and George" (Editorial Statement), *Electronic Literature Collection* 3 (February, 2016), Accessed February 7, 2017, http://collection.eliterature.org/3/work.html?work=fred-and-george.

60 Flourish Klink, "Fred and George" (Author's Statement), *Electronic Literature Collection* 3 (February, 2016), Accessed February 7, 2017, http://collection.eliterature.org/3/work.html?work=fred-and-george.

61 Montfort, *Taroko Gorge*, source code (taroko-gorge.html), lines 94–123.

62 Rettberg, "Tokyo Garage" (Author's Statement).

63 "Taroko Gorge Remixes."

Variability and Elements of Digital Literature

There are divisions, like the aforementioned separation into source and data or process- and data-intensive programs, that can be utilised in conjunction with both bots and other types of electronic literature. In his article "Five Elements of Digital Literature," Wardrip-Fruin offers models for understanding the different types of computation that might occur in works of electronic literature as well as the different elements to analyse when reading such works. Firstly, Wardrip-Fruin draws a distinction between works "for which computation is required only in the authoring process" and "those for which it is also required during the time of reception by the audience."[64] This division separates works that are presented through the medium of computing devices, including videos, websites, emails and any other electronic presentation, from works that are created with a computer and stored in a more persistent medium like physical books. It is worth questioning whether this first division is useful in this day and age. Traditional narrative books are published both as physical codices as well as e-books that, by definition, are computer-mediated. Social media posts, blogging and web journalism require computing both in the composition and viewing but are not generative in their nature. Computers play such a great role in all aspects of creating art that this division provides no aid in separating works into categories useful for analysis.

The second distinction described by Wardrip-Fruin is the aforementioned separation of works into computationally variable and computationally fixed works. Furthermore, the computationally variable works can be further separated into different categories based on the source of the variability: the works that "vary without input from outside the work's material" are 'batch-mode variable' while works that change based on external inputs are 'interactively variable.'[65] Interactively variable works can change based on a wealth of different data streams like the stock market, news, weather or user interaction. While the aforementioned distinctions are mutually exclusive, interactively variable works can be further classified into "those that vary with input [...] other than from humans" and "those that vary with input [...] from humans aware of the work," namely 'environmentally interactive' and 'audience interactive' works, of which one or both can describe an interactive work.[66] *Listen to Wikipedia*[67], a web page built by Stephen LaPorte and Mahmoud Hashemi, is a good example of an environmentally interactive work that transforms an external data stream into its artistic interpretation. It follows the data feed of recently changed pages on Wikipedia and turns the changes into a soundscape of bells and string plucks based on the amount of changed data and whether the modifier of the page is a human or an automated bot account. The work not only provides a visualisation and sonification of the relentless edits of Wikipedia pages but forces the viewer to think of the magnitude and the ever-changing nature of the whole open source encyclopaedia. *Stir Fry Texts* by Jim Andrews mentioned earlier can be seen as an example of an audience interactive work as it requires user's actions to generate new texts on the screen. In Andrews' work, the interactivity allows the user to read the generated texts in peace before choosing to change the text by sweeping over the screen with the cursor. As it can be seen, the source of the interactivity affects both the meaning assigned to the work and the reading experience at a very elementary level.

---

64 Wardrip-Fruin, "Five Elements," 40.

65 Ibid., 41.

66 Ibid.

67 Stephen LaPorte and Mahmoud Hashemi, *Listen to Wikipedia*, 2013, accessed March 21, 2017, http://listen.hatnote.com.

In his article, Wardrip-Fruin criticises Espen J. Aarseth's theory of cybertexts for focusing only on the traversal functions, the actual processes that combine textons into scriptons, "For example, works of digital literature carry out many processes—such as those determining the simulated emotional state of a virtual character—that are important to the literary functions but are not traversal functions for revealing or generating textons from scriptons [sic] (or can only be considered as such quite circuitously)."[68] While Aarseth divides his model of reading cybertexts into three parts, the operator (the reader), the verbal sign (the data) and the medium,[69] Wardrip-Fruin expands the model to include five elements to which attention should be paid while reading electronic literature. These elements include the aforementioned data and processes as well as interaction as described above.

The final two elements, surface and context, are somewhat intertwined. Wardrip-Fruin defines surface as "what the audience experiences: the output of the processes operating on the data, in the context of the physical hardware and setting, through which any audience interaction takes place."[70] The surface can be the website, the command line or the social media application, for example. The definition of context is more convoluted, the physical setting of the computer hardware being presented already in the definition of surface,

> Once there is a work and an audience, there is always context so this isn't optional. Context is important for interpreting any work, but digital literature calls us to consider types of context (e.g. intra-audience communication and relationships in an MMO fiction) that print-based literature has had to confront less often.[71]

The context can range from the choice of email reading software into the connections between audience members, which today might be more related to social media connections than the discussions between the users of Multi-User Dungeons. Another scholar, N. Katherine Hayles, also calls for paying attention to the context of the work and the connections between the surface and the context in her criticism of Aarseth's theory, noting, in her analysis of a website with a collection of digitised manuscript scrolls, that changing the work's surface, the way the audience encounters the work, also changes the meaning of the work,

> A moment's thought suffices to show that the changing of the navigational apparatus of a work changes the work. Translating the words on a scroll into a codex book, for example, radically alters how one encounters the work; by changing *how* the work means, such a move alters *what* it means. One of the insights into electronic textuality makes inescapably clear is that navigational functionalities are not merely ways to access the work but part of a work's signifying structure.[72]

Thus, not only should the software and hardware presentation of the electronic medium be under scrutiny when reading works of electronic literature, one should also consider how much the specific form affects the reading of the work and what it can tell of the greater context in which the audience encounters the work. For example, when considering Montfort's works, the context

---

68 Wardrip-Fruin, "Five Elements," 47.

69 Aarseth, *Cybertext*, 21. Wardrip-Fruin, "Five Elements," 47.

70 Wardrip-Fruin, "Five Elements," 48.

71 Ibid.

72 Hayles, *My Mother Was a Computer*, 90–91.

at which the works are presented, being shared online with a licence that allows remixing for any possible use, ties the context of the work into open-source software movement, which forms an interesting parallel to the tropes recycled and reinvented in literature.

Naturally, there are plenty of other things to consider when reading electronic literature. With web-based works like the ones described above, there are plenty of aesthetic and web-specific considerations to attend to. These considerations include, as noted by Glazier in *Digital Poetics*, compositional considerations, images, fonts, gifs, the structure of the URLs, the informativeness of the links, auto-refresh, time-based actions, interactivity, client-side programming, meta-tags and so forth.[73] As the approaches described in this chapter will be used in the following chapters to analyse bots and botmaking, with which both the aesthetic and the environmental factors differ considerably from other types of electronic literature, these compositional considerations will not be discussed in any greater detail here.

The Devil's in the Process

Although there are plenty of aspects to be considered when reading and analysing electronic literature, the aspect that is evident in most theories of electronic literature is the understanding of the work as a process instead of the resulting text that the reader encounters. This ties current works of electronic literature together with the works of procedural writing discussed in the previous chapter, as the understanding of the processes with which the work is generated is key to understanding both digital and non-digital procedural works. When reading works of electronic literature, like the varied oeuvre of Montfort's, we should not only read the surface text that appears on our screens, we should also open and examine the source code—its comments, algorithms and data. With some works like *Sea and Spar Between* and *House of Leaves of Grass*, the commentary can serve as a rationale, an artist's statement for the choices made in the creation process. However, also an unannotated source code can be elucidating even to a reader with no knowledge of programming languages: the plain text, meant to be processed and displayed on the screen, often jumps out from the lines of programming code. Being able to see the list of alternate phrases meant to flicker on the screen in *Flow My Tears* can make the reader of the work more attuned to other authorial choices, even when one is not able to discern the exact rate of refreshment from the source code. The myriad of remixes of the open-source licensed *Taroko Gorge* lead the reader of the different versions (and especially the reader of the licence included in the source code) to consider the openness of creative practices and software, pointing out that writing electronic literature can be something akin to, as Glazier points out, reciting the older tropes of oral poetry.

The rest of the chapters will focus on reading and analysing Twitter bots as electronic literature, taking into consideration the theories of Bootz, Aarseth, Wardrip-Fruin and others presented above. Furthermore, a lot of attention will be paid to understanding these theories together with botmakers' personal notions about their practice as seen in their interviews, blog posts and lecture notes.

---

[73] Glazier, *Digital Poetics*, 115–118.

4. Different Types of Bots

Generative bots on Twitter are a text evolving in time, appearing on the reader's timeline sporadically or at specific intervals, throughout the day, every single day. A bot can be read in multiple ways: One can follow the account to encounter its updates as they are rolled out by the program, placed between the updates of all the other accounts one is following—humans, brands and bots alike. One can read the archive of tweets on the individual bots profile page, most likely encountering thousands and thousands of more or less repetitive textual fragments. The third way, the viral way in which some bots reach a large audience, is by the way of retweets and quoted tweets. Retweeting on Twitter means sharing another account's update to one's own followers, which expands the reach of the individual update also to users not following the account that posted the original update. A quoted tweet is essentially the same thing, sharing an update of another account to one's own followers, but with added commentary. Followers of bot accounts can have very different reasons for sharing a bot's update as retweets or quoted tweets and these reasons, together with an analysis of what value they add to encountering bots, will be examined later in this chapter.

Twitter, as a platform, is opaque, making no clear difference between human writers and bot accounts on the profile pages of the accounts. Most artistically-geared bot accounts do announce that they post updates generated by a piece of software, crediting the author in, for example, the biography section of the profile. However, if not otherwise noted, the method of posting included in the metadata of individual tweets might possibly reveal the machine-created nature of the posts. As Twitter is fairly lenient with bot accounts,[1] there are plenty of bots created for many other purposes in addition to the artistic bots. There are spam accounts that post machine-generated content to generate ad revenue. Some bot accounts provide automated information about the service level of websites and public transportation. There is also one well-known case of human authors pretending to be a bot. The Twitter account *@horse_ebooks* was run by artists Jacob Bakkila and Thomas Bender for a couple of years, posting 'relatable' content thought to be generated by a machine. The account originally posted spam advertising for e-books but was later purchased by the artists. It gained a lot of traction and caused a stir in its audience when it was revealed that the account was actually part of a larger artwork by Bakkila and Bender and the tweets written by the artists themselves.[2]

Although there are several ways to encounter a bot's updates on the network, collected together or shared individually, the main question in any of these ways is how the updates generated by the bot's procedures are read and how they embody the motivations of their creators. How varied are the different intentions behind creating bots? What technologies do different types

---

1 Kazemi, "I, Twitter Bot."

2 Susan Orlean, "Man and Machine," *The New Yorker*, February 10, 2014, accessed January 10, 2017, http://www.newyorker.com/magazine/2014/02/10/man-and-machine-2.

of bots use? This chapter will focus on determining what makes bots different from posting 'hand-written' human-sourced updates on social media. Artistically-motivated bots will also be categorised into a few types based on the material they work with and the different processes they utilise.

Why Bots?

As noted already in the introduction, a maker of some of the best-known Twitter bots, poet, coder and educator Allison Parrish defines bots through their output. The content posted by bots is "repeated [...] / ... generated by a procedure / ... occurring over an extended (possibly infinite) period of time / ... embedded in an otherwise intention-typical context (i.e. non-procedural writing)."[3] She sees bots as an artistic way to rebel against the business logic of Twitter by subverting the platform's commercially sanctioned range of activities by utilising it for other purposes, much like graffiti-painters and skateboarders utilise the city and its public space in an alternative way. Although interventions on social media can be hand-written by human writers, Parrish notes that "interventions in real-time media *require* responses that repeat" for two reasons: Firstly, as "ad targeting / A/B testing of headlines / 'trending' topics / algorithmically curated feeds / spell-check, autosuggest and autocomplete" fill public online spaces with real-time computer-generated texts so interventions in the space also have to repeat in real-time to be effective.[4] As there are spam bots, bots attempting to sway political opinions and spread fake news and advertising bots constantly pushing their generated content into the network with no rest, the only level at which their messages can be subverted is working in the same non-stop cycle as they do. Secondly, as can be seen from the proliferation of all the aforementioned ad, spam and phishing bots, the technology for algorithmic creation and posting of content online is not only extremely cheap, it is actual the "lowest-cost, most practical solution to the problem—which is why even non-programmers seek them [automated bots] out."[5] In short, bots are a cheap and relatively easy way to create long-lasting interventions in the otherwise intention-typical, non-creative public space of continuously refreshed timelines where material older than a few hours is rarely viewed again.

Distinctions Between Bots

As the aim of this research is to focus on the procedural and generative aspects of bots and their relation to earlier procedural writing and electronic literature, a complete survey or classification of different kinds of bots is unnecessary and outside the scope of this research. However, since the rest of this chapter as well as the ones that follow will many a time make references to and comparisons between a few types of bots, a somewhat incomplete typology of bots needs to be drafted. Below is a proposal for distinctions by which different types of creative bot accounts can be classified and distinguished for the purposes of this analysis.

---

3 Parrish, "Understanding Bots."

4 Ibid.

5 Ibid.

Firstly, most common bots are certainly fully text-based, meaning their posts on Twitter appear as regular text updates. However, there exist plenty of bots which include media, images or video: Bots can post images chosen from a collection of open-source images, like the *Signe Brander Bot* (*@signebrander*) does with the public domain photo collection owned by the Helsinki City Museum. The content can be fetched from online sources according to some specific parameters. The *Freeze Frame Bot* (*@freezeframebot*) does this by fetching a short clip of a randomly selected YouTube video and adding a voiceover "Yup, that's me, you're probably wondering how I ended up in that situation," following a TV trope popular on Twitter.[6] Images can be wholly generated by a computerised procedure, like on the account *Tiny Spires* (*@tinyspires*) which posts algorithmically generated cityscapes inspired by Mary Blair's art. Text and images can also be paired in multiple ways. An image can be paired with a text update, like on *The Ephemerides* account (*@the_ephemerides*), the updates of which contain an image from NASA's outer planet probes together with a computer-generated space-alluding poem. The account *Tiny Space Adventure* (*@TinyAdv*), on the other hand, includes its text in the image, generating imaginary space shuttles complete with generated names and descriptions of them. Furthermore, there is a whole genre of bots that create images using the full set of Unicode characters, including the nearly 2,000 emoji symbols. For example, the *Tiny Star Fields* bot (*@tiny_star_field*) posts spaced out collections of symbols that represent or resemble stars to create night sky -like textures on the followers' timelines.

Another important division in the field of creative bots is based on how the content of the bot's posts is generated. To evoke Wardrip-Fruin's distinctions of electronic literature presented in the previous chapter, the generative bots that run based on their own internal logic are batch-mode variable while bots that procedurally work through sets of data are closer to computationally fixed works. An example of a generative bot that creates updates based on internal grammar rules and lists of words is Chris and Ali Rodley's *Magic Realism Bot* (*@MagicRealismBot*), which posts literary micronarratives in a variety of genres. A popular, now-defunct bot account that would be computationally fixed is Allison Parrish's *Everyword* bot (*@everyword*), which tweeted all the words in the English dictionary in the years 2007–2014. There are also plenty of big data bots that could be treated as interactively variable using Wardrip-Fruin's terminology. These bots are often journalistic or political in nature but they can also take an artistic approach in their funnelling of data into the textual form of Twitter. An example of this appropriation of real-world data for the purpose of entertainment or art is Darius Kazemi's *Two Headlines* bot (*@TwoHeadlines*), which combines headlines in real-time from Google News feed to form amusing or scary scenarios[7] like "Amazon Echo Arrested in Emissions Scandal"[8] or "Ryan Gosling says can test-launch ICBM at any time: official news agency."[9] There is also another interactive type of bot, which follows other users, bots and humans alike, and retweets or reposts[10] updates that

---

6 Mark Molloy, "*Record Scratch* *Freeze Frame*: The Hilarious Meme Taking Over the internet," *The Telegraph*, September 2, 2016, accessed January 12, 2017, http://www.telegraph.co.uk/news/2016/09/02/record-scratch-freeze-frame-the-hilarious-meme-taking-over-the-i/.

7 Chris Rodley and Andrew Burrell, "On the Art of Writing with Data," in *The Future of Writing*, ed. John Potts (Basignstoke: Palgrave Macmillan, 2014), 81–82, doi: 10.1057/9781137440402.0012.

8 https://twitter.com/TwoHeadlines/status/818444227357110272, accessed January 12, 2017.

9 https://twitter.com/TwoHeadlines/status/818308334663139328, accessed January 12, 2017.

10 The difference between retweeting and reposting being that the former utilises the functionality of the social network, preserving the original metadata of the post, including the user's profile

conform to the rules set forth in the bot's program. These bots often search for specific keywords, hashtags or rhyming schemes, like the *Pentametron* bot (*@pentametron*) which looks for the pentameter in people's posts.

Sources for Generation

Tiles and Grammars

One model of text creation that works particularly well with the short-form updates of Twitter, limited to 140 characters, is combinatory generation used already in Christopher Strachey's love letter generator. This so-called grammar-, tile- or tessera-based generation works with two components: the boilerplate grammar, which is a collection of fill-in-the-blank-type structures, such as different types of sentences, and a collection of words which are randomly chosen to fill the grammar structures. The grammar can be very complicated in its nature, consisting of a large number of structures the blanks of which can be filled by smaller structures combined from individual words.

There are plenty of examples of grammar-based bots, one of which is Nora Reed's *Thinkpiece Bot* (*@thinkpiecebot*), which parodies thinkpiece headlines of popular media outlets to "call out the predictability of these articles" and uses humour as "a way to cope with the fact that people keep writing them and keep defining my generation by the trumped-up bullshit in them."[11] The *Thinkpiece Bot* produces output as varied as "Has Kylo Ren Gone Out Of Style?"[12], "What Can Star Wars Tell Us About Sexting?"[13] and "I Voted For Hillary. Now I'm Going To Write Nazi Propaganda."[14] Two things make it easy for the reader to mistake the bot's posts for ones written by a human. Firstly, the direct references to politics, public figures and popular culture keep the themes of the bot current with what is going on in the media, which is something one does not expect from a grammar-based bot. Secondly, it is nearly impossible to see identical sentence structures being used for posts. The first aspect shows that the bot's vocabulary is constantly updated while the second suggests an extensive, highly-crafted grammar working in the background. Reed explains the structure of the grammar in their FAQ article on the bot,

> Each bot has a series of formulas that it picks at random and inserts words from predetermined lists. @Thinkpiecebot actually has two levels of these: the main formulas, such as 'Do [GENERATIONAL GROUP] Really Love [RANDOM WORD/PHRASE SELECTED FROM ANY CATEGORY]?', and a top-level formula that puts a publication prefix in front of one in six tweets,

Reed gives the update "Could Magnets Cure Gluten Intolerance?"[15] as an example and continues,

---

information, while the latter removes all the metadata and posts a copy of the update as a completely new update.

11 Reed, "The Official @Thinkpiecebot FAQ."

12 https://twitter.com/thinkpiecebot/status/820091246542733313, accessed January 15, 2017.

13 https://twitter.com/thinkpiecebot/status/820317736962244610, accessed January 15, 2017.

14 https://twitter.com/thinkpiecebot/status/820544229168529408, accessed January 15, 2017.

15 https://twitter.com/thinkpiecebot/status/640406837372825600, accessed January 15, 2017.

> As of this writing, @thinkpiecebot has main formulas and 25 variables. Some of these variables do not include very many options: the formula that created the above tweet grabs the verb—'cure'—from a list with only two available options, 'cure' and 'cause'.[16]

The nested variability in the grammar as well as the clearly oft-updated corpus of concepts both make the updates evolve in time, giving the illusion of human-sourced text. However, the relentless pace of the bot's hourly updates as well as the target of the parody clearly support Parrish's proposal that the best tool for fighting against the flood of painful headlines spun by media outlets is by splicing computer-generated humorous updates between the predictable, real thinkpiece headlines written by humans that often appear on almost any Twitter user's timeline.

Grammar-based structures do not merely work with textual outputs: the aforementioned *@tiny_star_field* bot along with many other emoji-based bots utilise grammars to generate their 'images' from the individual symbols. Grammars are also used at the code level to generate, for instance, images in the SVG format to be embedded in tweets. This method of generating images is included in the *Cheap Bots, Done Quick!* web service, which enables non-coders to easily create grammar-based text and image bots, handling the logistics of timed outputs to Twitter on the creators' behalf. A good example of generated images is the account *Vennsplain* (*@vennsplain*), the unpolished aesthetics of which resemble memes circulated on social media and imageboards. The *Vennsplain* account posts hourly updates with Venn diagrams of combinations of relatively abstract concepts hyped in contemporary media. Diagrams labelled with concepts like fakery and money with the label 'my happy place' at the intersection[17] and hype and notoriety with 'you are here' at the intersection[18] are often amusing at the face value but also propose a starting point for the reader to think about the connections between the disparate concepts on their own.

There are several reasons why grammar-based solutions are popular in creating Twitter bots. Firstly, randomness in text works only in pre-defined structures as "[r]andomly selected strings of words or musical notes don't have enough structure to make interesting meaning."[19] The limitation of very short updates enables the use of randomly populated boilerplate text because in a shorter narrative text there is not as high a need for coherence within the text as there is in a longer text; when generating a longer text, say, a short story, the reader of the story expects the details of the characters and settings to remain constant throughout the text. As botmaker Chris Rodley notes, "Twitter maybe isn't particularly amenable to long-form narrative, but it can do microstories and particularly premises or bloglines well. (See also Nora Reed's @thinkpiecebot)."[20] The short format of Twitter updates does not allow complicated narratives, making it a perfect pair to the grammar-based method, which also works most efficiently with a shorter text.

---

16 Reed, "The Official @Thinkpiecebot FAQ."

17 https://twitter.com/vennsplain/status/820615518285611008, accessed January 15, 2017.

18 https://twitter.com/vennsplain/status/820524923957420032, accessed January 15, 2017.

19 Kate Compton, "So You Want to Build a Generator," *Kate Compton* (Tumblr blog), February 22, 2016, accessed October 2, 2016, http://galaxykate0.tumblr.com/post/139774965871/so-you-want-to-build-a-generator.

20 Rodley and Rodley, "In Conversation."

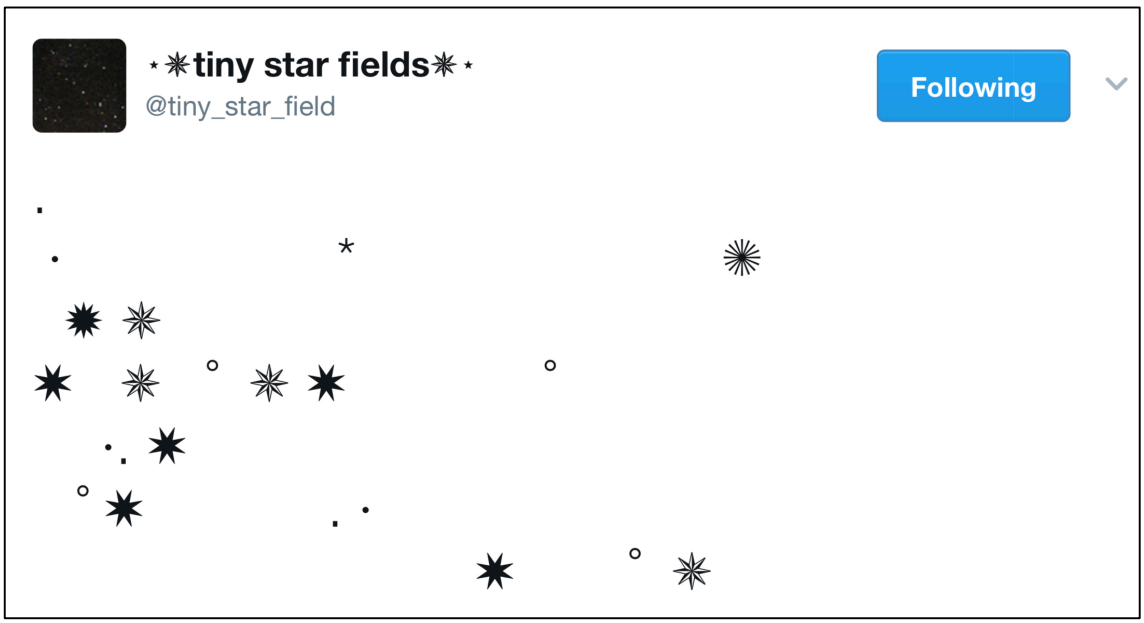

Figure 6. A screenshot of an update from the Tiny Star Fields *bot.*
(https://twitter.com/tiny_star_field/status/855866049073471493, accessed April 23, 2017.)

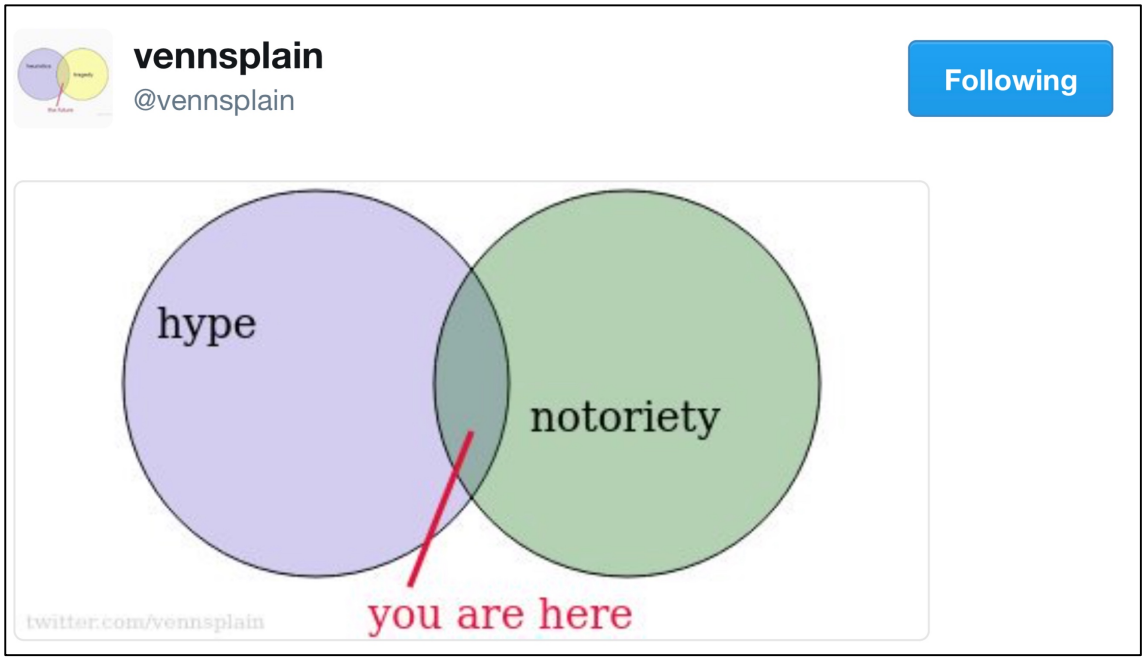

Figure 5. A screenshot of an update on the @vennsplain *account.*
(https://twitter.com/vennsplain/status/820524923957420032, accessed January 15, 2017.)

Markov Chains

Another common method of automatically generating text used both in bots and generators of longer texts is Markov chains. Markov chains are a tool of statistical analysis that works with text and other types of data. It calculates probabilities for combinations of words from an existing text and creates a new text based on the probability of word occurrences in it. As the statistical model does not consider the content of the input data, the readability and correctness of the

output text depend greatly on the variables chosen for the statistical analysis.[21] A longer source text is desirable as it provides more word combinations and enables the calculations to be more accurately representative of the source language, allowing for more variability and grammatical accuracy in the rehashed output. Thus, Markov-based text generators usually use whole books or literary corpuses, like all the public domain works available of Project Gutenberg, as their source text.

The so-called '_ebooks' accounts are a common trope in the bot scene and they utilise the Markov method. Named after the *@horse_ ebooks* account, for its style of sensible, interesting, yet garbled up updates, these accounts use any Twitter account's updates as their source text to construct updates that can be strikingly similar in style to the originals without being direct copies of them. When creating Markov-based bots, the choice of the source text becomes extremely important as the process itself is fairly standardised, offering few choices and variables to guide the process of analysing the text. Much of the entertainment value of Markov bots rises from the uncanny feel of facing a text that repeats the most common stylistic details of a human writer without achieving all the accuracy or the authenticity of the original source. The Markov method can be used to expose the formulaic nature of some specific way of writing as the algorithm is able to produce fairly legible artist's statements or impersonate another Twitter user's most typical tweeting styles.

Discoveries from the Familiar

There are plenty of entertaining and creative bots that are not generative in nature. Their only task is to publish updates from a predetermined list at regular intervals. Although the updates could be human-written prior to their scheduled posting, this genre of bots is generally focused on tasks too laborious and mindless for a human to complete, characterised by an "exhaustive sensibility," defined by Mark Sample as "the idea that while a human cannot tweet every word or every unicode character, a machine can."[22] The word 'list-based' will be used to describe this type of scheduled bots from here onwards. The prime example of a bot of this genre is the aforementioned *Everyword* bot by Allison Parrish. After the bot completed its task of posting every word in the English language, tens of remixes of the bot have been published: there are *Everyword*-type bots in other languages[23], going through numbers, hexadecimal colours and Unicode characters[24] and adding modifiers to the words to say that every word "is gay" or "fuck every word."[25] Part of the interest in these types of bots comes from gaining a possibility of reading through something so extensive that would normally require extreme dedication to complete. Also bots that go through the Bible or classic literature[26] one 140-character snippet at a time clearly fall into this category as they allow their followers to read through extremely long texts over an extended period of time.

---

[21] Cramer, *Words Made Flesh*, 74–75.

[22] Sample, "A Protest Bot."

[23] E.g. *Every Finnish Word* (*@kaikkisanat*).

[24] E.g. *Every Finnish Number* (*@EveryFinnishNo*), *Every Color* (*@everycolorbot*), *Everyunicode* (*@everyunicode*).

[25] E.g. *Every word is gay* (*@everywordisgay*), *Fuckeveryword* (*@fuckeveryword*).

[26] E.g. *Se Wsi Testamentti* (*@testamenttwit*), *Finnegans Wake* (*@finnegansreader*), *Ulysses Reader* (*@UlyssesReader*), *Willy Shakes* (*@IAM_SHAKESPEARE*).

In addition to the remakes of the *Everyword* project, there are other list-based bots that do not deal with dictionaries: For instance, there is a Finnish account *Edesmenneet* (*@EdesmenneetPrh*) that posts randomly selected names of inactive associations soon to be removed from the Finnish Register of Associations. This bot is based on a public domain list published by the Finnish Patent and Registration Office. In its case, the entertainment value comes from the absurdity or the historical resonance of the unknown and forgotten small associations, which include examples like "Real Relationships, Registered Association"[27], "Everything for Christ, Registered Association"[28] and "Youth Disco Society Alexander, Registered Association."[29] Followers of the account retweet the updates they deem most interesting, often adding a jocular remark about the name alongside the original post. I myself created a bot account *Kaikki on pahaa* (*@kaikkionpahaa*) which posted some thousand updates with the formula '[name of a food item] is bad' based on the official public domain list of food items and their nutritional values published by the National Institute for Health and Welfare in Finland. The followers of the account retweeted some of the funnier updates to their own followers, including the absurd "Exotic fruit is bad,"[30] often contradicting the original post by reminding their own followers that, for example, wine is certainly not bad.[31]

What unifies many list-based bots is that the content of the updates is very familiar and mundane—lists of different kinds of words, foods, colours and so forth—or is available in the public domain but consists of so many individual items that are impossible for a human to digest in their database format. While one can never be too certain about the range of material a grammar- or Markov-based bot produces, list-based bots have a well-defined scope. As the reader knows, often from the bot's name alone, the theme of the list through which the bot progresses, there is no surprise element in the type of the content posted on the bot's timeline. However, by spacing out the individual items of these extensive lists at regular intervals over an extended period of time, the bots allow the discovery of surprising details from familiar data: users might pay special attention to, for example, strange words in the dictionary, odd foods in open government data or emoji they have been unaware of before. This adds to the interestingness of the bots, keeping the reader following the account. Furthermore, individual words, colours or names might also remind the readers of moments, stories or jokes that they, in turn, wish to share to their own followers. In contrast to the formulaic grammar-based systems that risk being repetitive in the long run, list-based bots generally do not run into this problem as they go through each individual item on their list without repetition towards their evident, yet often distant, completion.

---

27 "Oikeat Ihmissuhteet ry." https://twitter.com/EdesmenneetPrh/status/800319036173258752 (Accessed January 15, 2017.)

28 "Kaikki Kristukselle r.y." https://twitter.com/EdesmenneetPrh/status/814165270944161793 (January 15, 2017.)

29 "Nuorisodiscoseura Alexander r.y." https://twitter.com/EdesmenneetPrh/status/820552359109623808 (Accessed January 15, 2017.)

30 "Eksoottinen hedelmä on pahaa." https://twitter.com/kaikkionpahaa/status/796882271575961602, accessed January 15, 2017.

31 "ei oo." Twitter user *@experiment769*, accessed March 27, 2017, https://twitter.com/experiment769/status/812021503281786880.

Lacking Distinctions

Placing an individual bot into one of these three categories is not straightforward at all times. While it is easy to categorise Markov bots due to their unpredictable output as well as the specific method with which their updates are produced, the distinction between grammar-based generative bots and list-based bots is sometimes hard to draw. One factor that separates the two is the determinability of the updates. Many list-based bots explicitly follow some system in their updates, be it an alphabetical list or following the sequence of a novel while posting 140-character snippets of a classic work. Grammar-based bots, mostly based on random selection of words and boilerplate text from the source, lack this determinability.

What makes a bot generative? There are plenty of *Everyword*-type bots which transform the word list used by the original *Everyword* account by placing the individual words into a simple boilerplate text like on the *@everywordisgay* and *@fuckeveryword* accounts referenced above. Technically, these accounts do utilise a grammar, albeit a very simple one without any variability in the sentence structure, and an extremely large word list, blurring the line between grammar-based and list-based bots. Furthermore, by adding modifiers to the list through the grammar— utilising the words in a specific sentence, for example— the bot account is able to generate stories, premises and interesting connections between concepts not present in the original data set. These kinds of fusion bots do not necessarily exhibit very clear list-like behaviour, either. An account titled *Gay Update* (*@GayUpdateBot*) works similarly to the *@everywordisgay* account although its vocabulary is limited to adverbs that are used to modify the boilerplate text "Gay Update: I'm currently feeling [adverb] gay." The placement of the adverbs into the sentence turns the list of words into narrative proposals which the followers of the account are able to share to their own followers. This account does not follow a particular order with its adverbs, thus reducing the predictability of its list-based nature. This also makes it harder to see the scale of the word list used: does it actually contain all the adverbs in the English dictionary or possibly just a smaller subset of them? This indeterminability of scale is more characteristic of grammar-based accounts.

Accounts that turn large amounts of data into readable narratives, working through a dataset row by row, also blur the lines between generative and iterative bots. There are accounts like *CensusAmericans* (*@censusAmericans*), created by Jia Zhang at *FiveThirtyEight*, which reconfigures the raw data from the US census into readable stories.[32] It produces updates like "I served between the Gulf War and the Vietnam Era. I work in beverage. I used to be on active duty. I am divorced. I got married in 1984."[33] To turn numbers and codes into sentences and micronarratives, the source of the bot must include boilerplate text and word corpus, much like any grammar-based bot. However, in this case, the trigger for the use of specific formulas is the actual source data instead of a random number generator. While this bot is not batch-mode random in its nature, it is certainly procedural. Furthermore, the account is also not artistically uninteresting by any standards, showing how difficult and arbitrary drawing lines between different types of bots can be. In the case of this bot, the reader should expect an accurate representation of the source data although the chosen approach into framing the text and stylistic choices do, of course, reflect the botmaker's attitudes towards the data.

---

[32] Jia Zhang, "Introducing censusAmericans, A Twitter Bot For America," *FiveThirtyEight*, July 24, 2015, accessed January 12, 2017, https://fivethirtyeight.com/datalab/introducing-censusamericans-a-twitter-bot-for-america/.

[33] https://twitter.com/censusAmericans/status/819317973605351425, accessed January 12, 2017.

However, there are also bots like Mark Sample's *NSA PRISM* bot (*@NSA_PRISMbot*), which is described as "an experiment in speculative surveillance."[34] This account simulates the large-scale data collection performed by the National Security Agency of the United States, as revealed in the leaks of 2013, by generating accounts of fictional people's online behaviour. The bot's updates identify fictional characters by name and they often include generated email subject lines, random locations and file names like "Bria Bogan of El Vinnie, Alabama saved a file called http://habbo.zip on Google Drive."[35] and "Laura Volkman of New Kellietown, Maine logged into Bing from 4.546° N, 33.676° E."[36] Furthermore, the bot "will occasionally flag (fake) social media activity using the list of keywords and search terms the Department of Homeland Security tracks on social media"[37] like "***FLAG*** @Rosie_Gibson mentioned 'ammonium nitrate' on Twitter. ***FLAG***"[38]

Both *@censusAmericans* and *@NSA_PRISMbot* clearly have an understanding of the enormity of data collection as their core idea despite the great difference in the way they function. The processes by which their updates are compiled certainly add to the meaning of the bot as it is hard to imagine a bot that spews out randomly generated census-like narratives to gain as much interest as one that does the same based on actual data. At the textual level, the reader might not see the difference so it is essential that the bots point out their generative method in the profile. It is necessary to know that *@censusAmericans* works with real-world data instead of randomly generated narratives and the account links to a blog post where the process is explained. In the case of the *@NSA_PRISMbot*, the actuality of the updates is secondary to the relentless pace of new plausible updates populating the reader's feed. As soon as the reader notices the word 'speculative' in the account's bio, it becomes clear that the continuing generation of imaginary flags on online behaviour is a way to point out "the way the individual invasions of privacy accumulate"[39] at the same time on some distant NSA servers. The *NSA PRISM* bot utilises the top selling point of grammar-based bots, the easy and cheap text generation, allowing the bot to oppose real-time data collection by the government in real-time, relentlessly and effectively.

The *@censusAmericans* and *@NSA_PRISMbot* accounts also expose another aspect that complicates dividing bots into specific types, the intent of the botmaker. Both of these bots can certainly be treated as artistic exercises or creative endeavours as they utilise polished, literary language to tell stories and to publicise grievances. This is true especially of *@censusAmericans* as the publication of anonymous narratives of individuals serves no clear practical purpose since the updates cannot even be used as a visualisation of the census data due to the enormous number of individual pieces of data and the impossibility of noticing trends at the level of individual updates. Yet, the *@censusAmericans* account is presented under the umbrella of a journalistic organisation, *FiveThirtyEight*, which focuses on data analysis, instead of a more artistic framework. The *NSA PRISM* bot, on the other hand, initially appears as a creative endeavour due to its "speculative" nature noted in its biography section, bringing to mind speculative fiction. However, in his article "A Protest Bot Is a Bot So Specific You Can't Mistake It for Bullshit," the bot's maker Mark Sample attempts to separate the bot from what he calls the 'bot canon' and its

---

[34] https://twitter.com/NSA_PRISMbot, accessed January 26, 2017.

[35] https://twitter.com/NSA_PRISMbot/status/723515829397151745, accessed January 26, 2017.

[36] https://twitter.com/NSA_PRISMbot/status/768302065432928260, accessed January 26, 2017.

[37] Sample, "A Protest Bot."

[38] https://twitter.com/NSA_PRISMbot/status/768138231179337728, accessed January 26, 2017.

[39] Sample, "A Protest Bot."

key markers "absurdism, comical juxtaposition, and an exhaustive sensibility."[40] The *NSA PRISM* bot, as well as "a bot that tweets the names of toxic chemicals found in contaminated drinking water or tweets civilian deaths from drone attacks," is "a bot of conviction" which "takes a stand" instead of a "*l'bot pour l'bot*," a canon-worthy bot, which might centre around the themes of "lost love or existential anguish," for example.[41] Protest bots have five characteristics which separate them from artistic, canonised bots: These bots are topical, data-based, cumulative, oppositional and uncanny. They are topical in that they are "about the morning news" and uncanny in that "they reveal something that was hidden" in plain sight.[42]

While useful in identifying protest bots in particular, the aforementioned characteristics can be generalised to aid the evaluation of other kinds of bots. Topicality in Sample's classification translates into the topic and the theme of the bot while the oppositionality of a bot can be applied more widely as the intent or the aim of the bot. These two aspects are linked in many ways: Artistic bots creating literary premises are often intended to be experiments in mechanical storytelling. Highly topical bots, like the *NSA_PRISM* bot, are often oppositional, aiming to expose and critique government wrongdoing. The *Thinkpiece Bot* is topical, oppositional and even uncanny, exposing the callousness of thinkpiece headlines while still providing its readers with sarcastic, comical and absurd updates. While all bots are cumulative, not all bots are data-based like the journalistic *@censusAmericans*. All the aforementioned cases show the blurriness between the categories of art, journalism, protest and even memes, although many bots do, indeed, follow the '*l'bot pour l'bot*' credo in that they are aesthetic, literary and/or entertaining while not sporting any other characteristics described by Sample than the cumulative method. In general, Sample's characteristics point out the most essential aspects to be considered about bots: when reading a bot, one should make note of the topic of the bot, the sources of its data and the aim for which the bot has been made.

Characteristics to Guide Analysis

All bots are repetitive, procedural and recurring over a long period of time. It is possible to attempt to categorise bots based by analysing their characteristics and methods. One way to understand the typical characteristics of a specific type of bots is by looking at the method by which their output is produced. However, as discussed above, this classification based on the method of generation is not clear-cut in any way: List-based bots placing adjectives into a single boilerplate sentence blur the distinction. Bots that interact with real-world data and turn this data into something journalistic or literary do not fit well into this categorisation, either. Furthermore, it is often hard to tell with certainty how a specific bot functions if the botmaker has not shared the source code in public or explained their design in a blog post or an interview. In these cases, the above categorisation might aid making an educated guess of the bot's method of generation based on the bot's output. Rather than seeing these types as genres or completely different kinds of works, this categorisation by the method can be helpful in the analysis as it can point out to the specific questions that should be considered when reading the bot's output.

---

[40] Ibid.

[41] Ibid.

[42] Ibid.

Furthermore, they can work as a framework to compare how the typical characteristics of a specific method of text generation are executed in a specific bot.

Another way in which bots differ greatly is their intent. There are 'canonised' bots that are artistic, literary and maybe humorous while some bots are bots of conviction, attempting to make a point. Other bots are journalistic, transforming and visualising data and trends into a format that is more easily digestible by readers. Even if the botmaker does not provide a rationale for their bot, it is possible to use the characteristics presented by Sample to analyse the intentions of the botmaker. The chosen topic tells a lot about the purpose of the bot; a bot can make poems, provide self-care reminders or it can be about the news, for example. The characteristics are also interrelated: a bot related to politics or the society, even if in a humorous way, can be highly oppositional, while also shedding light on issues kept hidden in the community. The source and use of the bot's data should also be evaluated. What is more, as shown by the comparison between *@censusAmericans* and *@NSA_PRISMbot*, it is essential to consider how the chosen method and the selected data support the topic, context and the intent of the bot.

Instead of looking for boxes into which a specific bot can be placed, the aspects presented above should be viewed as characteristics that can be used to analyse and compare bots. While Sample's characteristics can be helpful in analysing the most essential characteristics of any type of bot, making an educated guess about the method of textual generation based on the bot's output can provide help to understand the artistic choices, characteristics and metrics related to that specific method. The next chapter will present some of these choices related to grammar- and list-based bots in more detail.

5. Bot as a Black Box to Reverse-Engineer

Bots and Bootz

Bootz's model for reading works of electronic literature emphasises the importance of looking at the whole process of creation instead of the text-to-be-seen that can shift from one reading to another. Bootz, like many other theorists, points out that one should look at the source code and processes that make the text available to the reader. Even though there are plenty of works throughout the history of electronic literature that have not provided easy access to the work's source code and processes, like the proprietary hypertext systems from the early days of interactive fiction or multimedia works presented on CD-ROMs, many other works, especially web works and command line programs, provide the ability to glimpse at the processes running behind the text-to-be-seen at the push of a 'reveal code' button. For example, from the JavaScript source code of Montfort's *Flow My Tears*, one can easily discern the artistic choices made in the timing of the work as well as the selection of alternate texts that appear on the reader's screen. The same is possible with the *Sea and Spar Between*, whose source code reveals the formulas and textual particles with which the textual generation happens, explaining them in the comment sections spliced between the lines of code.

What makes bots different from other types of web works is that instead of existing as independent web pages, they are made public through the intermediary of the social network. A bot lives on a server separate from the social network and merely sends the output of its algorithms to be shared publicly. Thus, the reader is unable to take a look at processes that spew out new iterations of the formulas and data designed by the bot's author. Although there are many botmakers who explain the designs of their most successful bots on their own websites or in interviews and although there are easy-to-use tools that also encourage the publication of the bot's source code, like *Cheap Bots Done Quick!*, plenty of bots come without any explanation of their methods or any credit pointing to their maker. The reader might not have access to any other information about the bot than its earlier posts and the information included in the biography section of the Twitter account.

In her ELO 2016 conference talk, Parrish describes the process of reading bots and their output as 'reverse-engineering' them, "[T]o read a bot is to ask, 'how does it work?'"[1] As bots are essentially formulaic systems of rules and repetitive in their nature, a great part of reading and understanding a bot is going through its updates "'to factor out the template' and better understand the procedure that underlies the bot."[2] While electronic literature theorists argue against a close reading of individual texts-to-be-seen as they are not necessarily representative of the whole work, this kind of going through the updates to discover a formula is more of a distant

[1] Parrish, "Understanding Bots."

[2] Ibid.

reading strategy, enabling the user to see the grander formulas at play in the production of the texts. The large number of samples in the text generator's output allows the reader to see the formulas in action and note the words and sections that change from one sample to another. This availability of samples is the polar opposite of the unique-reading poem described by Bootz in his article,

> Yet, if the generator, as in the case of the unique-reading poem, generates only one text-to-be-seen, without the possibility of a reinitialization, the reader has no way of discerning the difference between the constant elements and those that are calculated inside what he or she is reading.[3]

Furthermore, the exploration and discovery of these formulas can also form part of the charm of reading bots, enabling the reader to further appreciate the mechanistic pairings of disparate concepts that human writers would hardly put together. This discovery is one of the main aspects of interest in the texts produced by the *Magic Realism Bot*, as noted by the bot's author Chris Rodley, "Another reason why the bot is a good fit for generating magic realist premises is that is can create links between ideas and concepts that aren't normally connected, that are totally divergent, in a way that humans don't easily do."[4] It is suggested to read through works of generative electronic literature several times to discern differences in the output texts. The discovery of a bot's formulas serves the same purpose, giving the reader of a bot a wider understanding of the work's source and data while refraining from using material extrinsic to the work in the analysis.

There are limitations to this reverse-engineering of the processes that generate a bot's output, which can be seen from earlier examples of electronic literature. Although Strachey published his generated love letters in an academic journal together with an explanation of the process, scholars of the surviving outputs were for a long time drawn to the conclusion that the word 'love' was utterly lacking from the letters, leading them to an analysis of the text through the queer lens of love "that could not speak its name."[5] Later observation of Strachey's original notes and source code has proven that the program did include not only the word 'love' but its derivatives and the words were absent only due to the limited amount of sample output texts available for the scholars' analysis.[6] Another caveat of extrapolating the rules and formulas from the output text can be noted from the case of the *Policeman's Beard Is Half-Constructed* whose boilerplate texts included a lot of wordplay not originating from the program but Chamberlain himself, as noted earlier. Another case where a simple reading of the updates might lead to a conclusion that differs from the reality is the case of *@horse_ebooks*. However, the *@horse_ebooks* account was treated more like a Markov-based spambot whose updates do not replicate any particular formula, making the reverse-engineering of the bot's processes harder for the reader in the first place. Furthermore, the style of the account's updates deliberately attempted to emulate machine-like language and to keep up with the illusion of being programmatically generated.

The functional model of Bootz's, presented in chapter 3, can easily be adapted to the process of authoring a generative bot. The model can also elucidate several important elements of the work that should be focused on when reading and analysing bots in the context of electronic

---

3 Bootz, "The Functional Point of View," 312.

4 Rodley and Rodley, "In Conversation."

5 Wardrip-Fruin, "Five Elements," 33.

6 Ibid., 50.

literature. The writing of a generator of any kind, but especially a bot, differs greatly from other kinds of electronic literature, which are often longer in form and have a different level of interactivity or time-based changes. With generative bots, the author-text, the source written by the bot's creator, can consist of grammar, corpora and functions. The bot's grammar is the collection of rules and sentences from which the output is chosen. It also includes information on how to fill the sentence structures with words from the bot's corpora. Corpora—the plural of a corpus, "[t]he body of written or spoken material upon which a linguistic analysis is based"[7]—describes all the different lists of words created, collected or curated by the author from which the blank placeholders in the grammar are filled. Functions can be used to describe other functionality of the bot, which can, in the case of interactive bots, for example, be the handling of external real-time data feeds like weather information or the parsing of database information like census data. Both the grammar and the corpora can easily be discerned by the reader from the output text on the bot account's timeline, without necessarily needing to look at the source code. The functions of the bot might be harder to discover without access to either the actual source code or an explanation of the bot's connection to external data sources. This explanation, however, can be included in the bio section or even in the name of the bot. *@censusAmericans*, for example, makes it abundantly clear that it works with actual census data by noting "Tweeting the census one real american at a time"[8] and expanding this in a linked blog post on the *FiveThirtyEight* site.

Corpora

While the text posted by the bots might not be considered human-authored, all the three elements of the author-text of the bot, in addition to the original source code running the bot, are products of human creativity. The writing of a bot, or any other generator, differs greatly from writing interactive fiction or traditional literature. Chris Rodley, the co-creator of the *Magic Realism Bot* alongside Ali Rodley, describes the process of authoring their bot in terms of both writing and collecting, "Two of the key writing tasks were to create the bases or templates, and to collect a large vocabulary to populate them with."[9] The word 'curation' could be used to describe the collection of words to populate the different word lists required to fill the templates in the grammar. These corpora can come from the author or they might be hand-picked by the authors to fit with the theme of the bot, like in the case of *Magic Realism Bot*,

> Magic Realism Bot uses its own corpus and so doesn't fit into this category of data-driven writing; while we could have drawn upon large online word lists or Wikipedia, we wanted to be able to tightly control the vocabulary that was being outputted (we want a grandfather clock, say, but not a Swatch watch on our list of machines).[10]

---

7 *Oxford English Dictionary*, *OED Online*, s.v. "corpus," accessed January 27, http://www.oed.com/view/Entry/41873.

8 https://twitter.com/censusamericans, accessed January 27, 2017.

9 Rodley and Rodley, "In Conversation."

10 Chris Rodley, "Introducing Chris Rodley," interview by Sarah Blaszczok, *Abandon Normal Devices* (blog), April 12, 2016, accessed October 2, 2016, http://www.andfestival.org.uk/blog/introducing-chris-rodley-the-art-of-bots/.

As noted by Rodley, bots can also make use of a wealth of large-scale open-source data sources and words lists, like Wikipedia or its structured data version, Wikidata. There are also specific crowdsourced projects designed to provide specialised word lists for the making of "weird internet stuff," like Darius Kazemi's *Corpora* project, which includes crowdsourced topical word lists released into the public domain free of charge.[11] The *Corpora* project contains short lists of words with the interests of testing and quick creation in mind, "host[ing] a sampling of 1000 common nouns, adjectives, and verbs" in addition to specialised lists, including a list of US federal agencies, mass surveillance project names and Netflix film categories.[12] There are also more extensive application programming interfaces, like Wordnik, for example, through which bots can be connected to large-scale dictionaries.[13] While the *Magic Realism Bot* utilises hand-picked corpora to keep up with its theme, the *NSA PRISM* bot by Mark Sample is "rooted in specificity" in that "[t]he Internet companies the bot names are the same services identified on the infamous NSA PowerPoint slide. When Microsoft later changed the name of SkyDrive to OneDrive the bot even reflected that change."[14] In the case of the *NSA PRISM* bot, the adherence to the details of the NSA leaks adds to the power of the bot to represent possible, yet fictional tracking of people in its updates.

One of the selling points of bots is that they can handle extensive data and long word lists, making it is easy for botmakers to appropriate interesting lists and data from a variety of sources. This is particularly visible in the list-based bots mentioned in the previous chapter, which appropriate government data for aesthetic and entertainment purposes. However, longer word lists are also very prominent in grammar-based bots as more extensive corpora increase variation in a bot's output, keeping it interesting for a longer period of time.

Even word lists written by the authors can be seen as being collected from the surrounding cultural canon and context: For example, one of my own bots, *@soinismibot*, attempted to spoof the inventive neologisms published in the blog of the Finnish Minister of Foreign Affairs, Timo Soini. The blog contains a myriad of invented terms intended as insults to the ones opposing Minister Soini's politics. In the summer of 2016, his use of words like "bicycle communism,"[15] "stone therapy communism"[16] and other untranslatable neologisms was noted by the online supplement of *Helsingin Sanomat* newspaper[17] and quickly picked up in social media. The article inspired me to read through the last three years of the blog posts and to list words of a similar nature used in the blog earlier, finding terms like "straitjacket feminism,"[18] "bonehead atheism"[19]

---

11 Darius Kazemi, "Corpora," Github, September 23, 2016, accessed January 28, 2017, https://github.com/dariusk/corpora.

12 Ibid.

13 "About Wordnik," Wordnik, accessed January 28, 2017, http://www.wordnik.com/about. Kazemi, "Corpora."

14 Sample, "A Protest Bot."

15 "fillarikommunismi." Timo Soini, "Haastamalla horjutetaan – punavihreät kyykkäsivät," *Ploki* (blog), May 17, 2016, accessed March 28, 2017, http://timosoini.fi/2016/05/haastamalla-horjutetaan-punavihreat-kyykkasivat/.

16 "kiviterapiakommunismi." Timo Soini, "Lässytys lyö takaisin," *Ploki* (blog), June 5, 2016, accessed March 28, 2017, http://timosoini.fi/2016/06/lassytys-lyo-takaisin/.

17 "Ulkoministeri Soini kirjoitti Nytin olevan 'kiviterapiakommunismia', mitä se tarkoittaa?" *Nyt*, June 6, 2016, accessed March 28, 2017, http://www.hs.fi/nyt/art-2000002904869.html.

18 "pakkopaitafeminismi." Soini, "Haastamalla horjutetaan."

19 "luupääateismi." Timo Soini, "Olenko minä nero?" *Ploki* (blog), June 19, 2016, accessed March 28, 2017, http://timosoini.fi/2016/06/olenko-mina-nero/.

and "watercress communist."[20] Based on these notes, I built *@soinismibot*, which uses a simple template to combine a hippy-sounding plants or aspects of urban hipster life with political -isms. With a short list of some hundred prefixes and 75 suffixes, the bot produced neologisms like "soy dog despotism,"[21] "pulled oat elitism"[22] and "restaurant day opposition."[23] Although some of the word parts included in the corpora were actually sourced from Minister Soini's blog posts, I quickly expanded the lists to include concepts that I thought exemplify the lifestyle of young, urban leftist greens Minister Soini attempts to vilify in his blog. The words did not come directly from any single source but they were certainly curated to sample the self-ironic references to the lifestyle of young urban hipsters visible on my social media timelines. Plenty of my friends also chimed in with their own suggestions, which I also included in the corpora of the bot.

Nora Reed's critically-framed *@thinkpiecebot* follows a similar method of looking at the target of its criticism and appropriating phrases, structures and words from the headlines the bot aims to ridicule,

> Most of it is words and phrases I came up with while looking at horrible thinkpieces, but I got a lot of help from my Twitter followers. They did particularly invaluable work with helping me phrase some of the issues regarding marginalization and privilege; I wanted to be sure that wasn't falling into doing 'ironic bigotry', and they helped a lot with coming up with specific phrasings that wouldn't harm groups who are already being targeted by actual thinkpieces.[24]

Reed notes that curation and limiting of the corpora is important so that the bot does not end up repeating the hateful speech that it is intended to criticise, "However, there are a lot of places I don't want @thinkpiecebot to go because they end up way too close to just parroting the people the bot is meant to make fun of."[25]

The curation and collection of data and words from a wide range of sources raises questions that are also relevant in discussions of appropriation of texts into artistic use. When reading the output of bots, the reader should consider the words that are used to flesh out the template sentences: what kind of words are included, what is excluded and what kind of authorial attitude the word choices convey. We know that the corpora in *Magic Realism Bot* are chosen to exclude mundane and non-literary objects to keep up with the Magic Realist themes in its story premises. Similarly, it is evident in the updates of *@soinismibot* that the bot is intended to ridicule the lifestyle of young urban hipsters as much as Minister Soini's blogging. The *NSA PRISM* bot focuses on companies detailed in the NSA leaks to keep its updates and criticism believable and relevant.

---

[20] "vesikrassikommunisti." Timo Soini, "Politiikka on ihanaa," *Ploki* (blog), June 12, 2016, accessed March 28, 2017, http://timosoini.fi/2016/06/politiikka-on-ihanaa/.

[21] "soijanakkidespotismi." https://twitter.com/soinismibot/status/751783505689018368, accessed March 28, 2017.

[22] "nyhtökauraelitismi." https://twitter.com/soinismibot/status/749488365473497088, accessed March 28, 2017.

[23] "ravintolapäiväoppositio." https://twitter.com/soinismibot/status/750998347486486528, accessed March 28, 2017.

[24] Reed, "The Official @Thinkpiecebot FAQ."

[25] Ibid.

Grammars

"The most common way that generators fail is that they produce content that fails to be interesting,"[26] notes Kate Compton about the shortcomings of grammar-based generators in her post "So You Want to Build a Generator," continuing that achieving novelty with Twitter bots is difficult due to the number of updates a bot will publish during its runtime,

> Most generate multiples, but a twitterbot posting every hour will generate more content than a novel-generator outpouring one novel every NaNoGenMo. So achieving novelty with the first Twitterbot will be more difficult because there are so many artifacts being produced that any given one of them will probably start seeming less special.[27]

A bot with a simple grammar, with a small selection of sentence templates to choose from, just like a bot with a small selection of words, is more likely to become repetitive than a bot that has a wide variety of templates or words to choose from. However, even well-crafted bots with an extensive grammar can become repetitive in the long run, requiring regular maintenance from its creator. This is noted by many botmakers as they explain their process. For example, Chris Rodley notes in an interview, "As it has gone, we've progressively made these syntaxes much more complex and stochastic, because after a while those recurring templates get boring."[28] The *Thinkpiece Bot* not only requires maintenance to avoid being repetitive but Nora Reed notes that they keep updating the bot to stay current in its criticism,

> Thinkpiecebot's funniness in particular comes from the unexpected combinations that it produces being put into the recognizable headline format, but doing that ended up being complex– I have over 50 formulas in it and nearly as many variables, and I'm constantly updating it so that it keeps up with the zeitgeist.[29]

In her blog post, Kate Compton introduces the 10,000 Bowls of Oatmeal Problem. Many generators with very large output quantities, like bots, are prone to this problem,

> So your algorithm may generate 18,446,744,073,79,551,616 planets. They may each be subtly different, but as they [sic] player is exploring them rapidly, will they be perceived as different? I like to call this problem the 10,000 Bowls of Oatmeal problem. I can easily generate 10,000 bowls of plain oatmeal, with each oat being in a different position and different orientation, and mathematically speaking they will all be completely unique. But the user will likely just see a lot of oatmeal. Perceptual uniqueness is the real metric, and it's darn tough.[30]

The problem with generated output is that if the differences between individual items are subtle, they do not appear unique enough to the audience. Compton describes two metrics with which generated output as a whole can be evaluated, perceptual differentiation and perceptual uniqueness. While perceptual uniqueness requires each generated item to be distinct and

---

[26] Compton, "So You Want to Build a Generator."

[27] Ibid.

[28] Rodley and Rodley, "In Conversation."

[29] Nora Reed. "How and Why @nerdgarbagebot Works," *Gusty Winds May Exist* (blog), January 10, 2016, accessed October 2, 2016, http://barrl.net/2801.

[30] Compton, "So You Want to Build a Generator."

memorable as such, "[p]erceptual differentiation is the feeling that this piece of content is not identical to the last."[31] While perceptual uniqueness can hardly be expected even from bots with extensive grammars as words and templates are bound to repeat over time, perceptual differentiation is one metric with which the success of bots can certainly be evaluated based on looking at the outputs of the bot at large. The *@soinismibot* eventually failed with perceptual differentiation due to its use of a word list limited to some hundred particles, which, at the pace of one update per hour, appeared several times a day in the bot's output, often in two sequential updates. The bot gained a following of some 1,800 users in just a couple of days but the sharing and discussion of its output started to dry out as soon as the bot's first week was finished. The bot was retired, as suggested in private messages by a number of followers, after just two weeks of running as the updates did not remain distinctive enough to be interesting in the long run.

Botmakers are certainly intuitively aware of the need for perceptual differentiation between individual outputs, as they keep honing their code in order to achieve as varied results as possible from the same templates. The Rodleys have certainly kept perceptual differentiation in mind when making their templates as accommodating as possible to different categories of words,

> One principle we try to follow is to get as much variation as possible with each template; ideally, stories from the same template should look mostly or totally different. One way we do this is by making the vocabulary as flexible as possible, and employing as many categories of things we can at all times. Probably our two most fruitful and frequently used categories are simply 'concrete things' (like clocks and swans) and 'abstract things' (like love or capitalism). The variability means that even we are surprised by the results a lot of the time, despite knowing everything that's in the database.[32]

No Code Necessary

Perceptual differentiation is one good way of evaluating the design of the author-text of a bot without needing to look at the actual source code, grammar or corpora of the bot. Furthermore, by comparing the individual outputs on a bot's timeline with one another, the reader can gain an understanding of the variability of the templates and word lists used, even though the sources of this variability might not be individually categorisable as the result of extensive templating or large word lists, as can be understood from the innovative use of widely applicable templates in the *Magic Realism Bot*.

This method of evaluating the bot's author-text based on the mass of its outputs, by looking at the sources of its corpora, the variability of its templates and the perceptual differentiation of its outputs at large, allows also readers who are not particularly code-savvy to evaluate and critique the bot as electronic literature. Rather than requiring any knowledge of programming languages or the technologies used to make the bot work, this way of reverse-engineering the output of the bot at the conceptual level makes the reading of bots similar to reading any other work of combinatory or conceptual literature. If one can conceptually grasp the diversity of outputs made possible by the design of Queneau's *Cent mille milliards de poèmes*, one is perfectly capable of grasping the processes and concepts that lay behind the outputs of artistic, combinatory bots.

---

[31] Ibid.

[32] Rodley, "Introducing Chris Rodley."

Reverse-engineering a bot in the way Parrish suggests us to do is the same way Florian Cramer describes reading concrete poet Eugen Gomringer's permutational text "no error in the system" in which the phrase 'no error in the system' is repeated over and over again as the first occurrence of the letter 'e' shifts one position forward in each line,

> no error in the system
> no reror in the system
> no rreor in the system
> [...][33]

Cramer notes that "'no error in the system' is tautological in two respects: First of all the lines become redundant and repetitive as one has grasped the algorithm."[34] Much in the same way, when reading a bot, one attempts to grasp the combinatory algorithms, to look at the templates that remain the same while the words change, like one would do when looking at the texts produced by Queneau's book, Strachey's love letter generator or Knowles and Tenney's *House of Dust*. At this basic level of combinatory processes, understanding bots requires no knowledge of their networked or computerised nature and much less knowledge about the specific details that go into their programming. As the above analysis of the elements that make up the author-text of bots shows, the choice of corpora and the variability of the templates selected by the author is certainly more important that the actual source code that keeps the bots running.

The Reader's Field

Works of electronic literature question what we consider reading and writing to be, notes Stephanie Strickland in her 2006 article, "Writing and reading relations and behaviors have changed: 'writing' has addressed itself to producing behavior (executable code), and 'reading' has evolved to receive and participate in dynamic hypermedia."[35] With the writing process one step removed from the text-to-be-seen, the creation of the author-text consists of producing behaviour to be executed by the computer. Reading, in many cases, has become interactive and, as noted in chapter 3, there are several reasons why the read-text might differ from the text-to-be-seen. Furthermore, with bots, the reader's relation to the text-to-be-seen is very special, very interactive.

The context of bots is the everyday life. Bots are read and encountered during everyday activities, with the same devices people use for both business and pleasure. The context of bots is the web browser and the social network. Discussing the work *Dakota*[36] by Young-Hae Chang Heavy Industries, a work that can be categorised as an animated text, Noah Wardrip-Fruin makes notes about the nature of the network context, which is very useful in understanding the context of bots as well,

---

[33] Eugen Gomringer, "3 variationen zu kein fehler im system," in *Konkrete Poesie*, ed. Eugen Gomringer (Stuttgart: Reklam, 1972), 63–64, quoted in translation in Cramer, *Words Made Flesh*, 65.

[34] Cramer, *Words Made Flesh*, 65.

[35] Strickland, "Writing the Virtual."

[36] Young-Hae Chang Heavy Industries, *Dakota*, 2001, accessed April 23, 2017, http://www.yhchang.com/DAKOTA.html.

> There is something about the network, and about the growth of network culture (especially forums for posting, finding, sharing and rating works—from sites to particular animation aesthetics through the teeming heterogeneity of *YouTube*) that has been important in development of this work. And something about the ability to browse for and view this work in a web browser, using the same machine used for work, during any brief break from work.[37]

Bots exist as part of this network culture, posting their creations on a platform designed for both viewing but also sharing and commenting upon such content. There is no barrier for viewing the updates of bots, much like there is no barrier for viewing web works; the art is readily accessible on the devices used by people, for their brief reposes from work.

Reading Twitter bots and their outputs, like reading any updates on social media, is not a passive process. All bots are interactive in the way that users may share the bot's creations to their own followers, discuss them in real-time with other members of the community and even build upon the bot's updates. Some bots are also directly interactive in the way they work. They might respond to user's comments, like or retweet them or they might even allow users to request their own custom updates. For example, the Finnish *@jaxubotti*, making fun of overly positive social media updates, responds to users using specific hashtags or directly mentioning the bot. Another bot titled *Neon Clock Tweet Bot* (*@NixieBot*) responds to users using a specific hashtag with an animation of the words requested by the users being displayed on a neon clock screen.

Although Chris Rodley notes the importance of the retweet function in the organic growth of a bot's audience, "Twitter as a platform is great for bot writing, of course, because the retweet function helps 'successful hits' get more widely seen,"[38] the sharing of a bot's update also often adds value and meaning to it. This is particularly common with the *Everyword*-type bots, the posts of which are often quote-tweeted and shared to the reader's own followers with a joke, anecdote or a story. For example, my own bot, *Kaikki on pahaa*, posting a list of food items with the words "[food item] is bad" added, gained quite a lot of shares from its followers. For example, the post "Fortified wine is bad."[39] was shared with a note "But effective!"[40] and "Please be more specific"[41] was added to a retweet of "Salad is bad."[42]

Distributed Reading

Another aspect of reading that comes into play with list- and data-based bots is that the reading of the original data is distributed. The author of the bot will not have read all the data before it is outputted to the social network and neither will any individual reader read all the updates. N. Katherine Hayles gives an example of this so-called distributed reading when discussing another work of electronic literature, Jim Campbell's *I Have Never Read the Bible*. In Campbell's

---

37 Wardrip-Fruin, "Five Elements," 42.

38 Rodley and Rodley, "In Conversation."

39 "Väkevä viini on pahaa." https://twitter.com/kaikkionpahaa/status/812283755029295104, accessed January 30, 2017.

40 "Mutta tehokasta!" Twitter user *@amimnts*, accessed January 30, 2017, https://twitter.com/amimtns/status/812284411735633920.

41 "Tarkenna vähän." Twitter user *@Jirgson*, accessed January 30, 2017,https://twitter.com/jirgson/status/808738299569377280.

42 "Salaatti on pahaa." https://twitter.com/kaikkionpahaa/status/808735375485190145, accessed January 30, 2017.

installation, a synthesised voice emanates from a 19th century *Webster's Dictionary*, reading the Bible letter by letter. The voice is that of the artist reading each alphabet over Mozart's *Requiem*, which gets also gets remixed as it is fed through a computer algorithm playing the alphabet recordings in the sequence as they appear in *King James' Bible*.[43] Hayles describes the reading of the Bible in the installation as reading distributed between the artist, the algorithm and the audience,

> The point, after all, is that 'I Have Never Read the Bible,' that is, the artist as a singular subject has not read it. Rather, 'reading' here is a distributed activity taking place partly in the articulations of the artist, partly in the 'voiced' text, partly in the Oreo structures of the scanner, computer, and synthesizers, and partly in the perceptions of the viewer who not only makes words out of the voiced letters but also makes meaning out of her interpolation in to this distributed cognitive environment.[44]

Especially the meaning-making in 'this distributed cognitive environment' on the part of the audience is relevant to bots, as the same process occurs when reading works that are based on large amounts of data or text. In the case of *Everyword* bots or *@censusAmericans*, for example, the audience, when reading the bot's posts, is keenly aware of the amount of data behind the individual updates. The reader of an individual update knows that the posting of every word in the dictionary or the story of every single individual census participant will take years to complete. The reader is aware that the data being posted has certainly not been fully read through by the author or maybe any human being. This awareness of the source of the bot's data acutely points out that the continuous, non-stop updates like "I work for the state government. I don't have health insurance. I moved last year. I had less than 2 weeks off last year."[45] and "I went to college for less than a year. I work in landscaping services. I get to work around 6:45am. I have never been married."[46] on the *@censusAmericans* account cannot all be written by a human.

As the reader becomes familiar with a larger amount of *@censusAmericans* updates, they are able to see how formulaically the data points are turned into readable narratives. Based merely on the appearance of the individual updates, the reader becomes conscious of the technological process that is involved in turning a numerical database into readable text. Although they do not need to understand the server functions, databases or the actual source code of the bot, the individual updates are enough to make them understand how the process of writing is distributed between computers and the author, affecting also how they make sense of the updates and give them their own meaning and interpretation. Like Hayles notes about *I Have Never Read the Bible*, the meaning of an individual bot's updates also does not solely depend on their content. Rather, understanding the whole process involved in both collecting the original data and parsing it into a readable Twitter-sized format play a great part in reading list- and data-based bots. While the narratives generated by *@censusAmericans* might be interesting and engaging, inviting the reader to complete the stories in their mind, the appeal of the account is in the actual process that turns

---

43 Jim Campbell, "I Have Never Read The Bible," Jim Campbell, accessed February 6, 2017, http://www.jimcampbell.tv/portfolio/objects/memory_works/i_have_never_read_the_bible/. Hayles, *My Mother Was a Computer*, 212.

44 Hayles, *My Mother Was a Computer*, 213.

45 https://twitter.com/censusAmericans/status/828256873543561216, accessed February 6, 2017.

46 https://twitter.com/censusAmericans/status/828407867338276866, accessed February 6, 2017.

a huge amount of individual data points into readable—perhaps even literary or journalistic—narratives.

The distribution of work between the human author and the computer is pointed out by every bot and the awareness of this division not only flavours the reverse-engineering of list- and data-based bots but also that of the grammar-based generative bots. While the design of the author-text requires human creativity, the readers often ascribe the authorship of individual updates to the computer, as noted Rodley, "Many people seem to ascribe full authorship to the Twitter bot, which I don't mind, but I find a bit unusual, because I know how long it took to get it tweeting coherently. Actually, I think, it's not unusual now in digital writing to see authorship split between the primary creator, automated processes or bots, and increasingly other users on the Internet."[47] Furthermore, many botmakers do attempt to make their bot as varied in its output as a human author, "Currently, I think, many botmakers are focused more on the task of training bots to be as human as possible, rather than making them 'machiney'. Because humans are already pretty good writers, it's a tough task to make a bot that's as good or better than one. But what we can focus on is making a bot that's good at being a bot."[48] As the grammar of a bot becomes more varied and extensive, the more human-like and less formulaic the resulting updates become. This, of course, makes it more difficult for readers to reverse-engineer the process generating the updates, obscuring the origins of creativity in the botmaking process.

Bots as Ludic Dysfunction

As noted above, people encounter bots during their everyday activities with the same devices they use for both work and pleasure. Bots appear alongside updates from human writers, in a context that can be called intention-typical as most updates on social media are straightforward communication instead of more experimental and artistic uses of language. The mechanistic combinatory language of bots presents the readers with juxtapositions a human writer might not even be able to imagine, disrupting the reading of social media with language that does not particularly fit with the flow of everyday speech on people's timelines.

This disruption of non-typical language spliced between the regular updates from people, media outlets and brands functions in many ways. As noted about the *Thinkpiece Bot* earlier, the generation of updates that look similar to actual headlines can be a method of criticising the media for the predictability of their articles. Allison Parrish, on the other hand, claims that aesthetic bots of all kinds can act as resistance to the way Twitter expects the users to use the platform, much like graffiti or skateboarding question the *status quo* way of utilising public spaces in cities. In the notes of her ELO 2016 talk, she describes how Twitter monetises user behaviour,

> [U]ser A tweets about a brand, user B engages with a tweet — maybe favoriting it or retweeting it or replying to it. Twitter uses tweets like this to form demographic profiles, bundles of statistical features about their users that enable them to sell ads to brands. the ideal Twitter

[47] Rodley, "Introducing Chris Rodley."

[48] Ibid.

> user is a free source of textual labor, relating individuals to the brands that want to sell to them.[49]

Giving her own *Everyword* bot as an example, bots can disrupt this monetisation by engaging users with content that has no practical use for advertisers,

> [A]s of today, the word 'sorry' has been liked and retweeted thousands of times. this is an example of a kind of engagement that twitter can't monetize in a straightforward way. so 1300 people retweeted the word 'sorry'—who is going to buy ads based on that? board game manufacturers?[50]

Describing how skateboarding creates new meanings for objects and spaces in the city by turning the design of a simple handrail into an object to play with, Parrish suggests that one of the main functions of Twitter bots is "to turn twitter into something enjoyable, something that can be manipulated."[51] This process is the same as the process in which the purpose of the handrail, dictated from the top down by the designer of the object, gains new playful use or a space without meaning, like an empty parking lot or a school yard after school, gains a new meaning as it is used for skateboarding. Most of our public spaces are highly designed top-down and restricted in their use. The playful use of these spaces, the car parks, the social network, can fill them with a meaning that comes from the users and creators instead of the restrictive framework of the corporate designers. Bots can, Parrish continues, "take the barren landscape of brand hashtags and tweets about the oscars and create an eruption of meaning where twitter had been content to say nothing."[52] This meaning, being as varied as described earlier, offers aesthetic value and discovery to the readers as well as enables a real-time critique of real-time media.

Marie-Laure Ryan calls this ludic, playful, dysfunction, "Whereas political dysfunctionality asks: how can I subvert this technology to encourage critical thinking, ludic dysfunctionality grows out of the question: what can I do with this technology, other than what it was meant for?"[53] The presentation of computer-generated text amongst intention-typical updates by people and brands already utilises the social network in something it was not meant for. Furthermore, the content produced by bots also subverts Twitter's monetisation model by making the users like and share content not useful in building profiles for advertising purposes. Ryan readily notes short fiction on Twitter as an example of ludic dysfunction in her article,

> In the same spirit, technologies can be deprived of their primary function and turned into art toys: for instance, the limited graphic capabilities of the ASCII code have been used to produce images; poetry has been created out of computer languages; short fiction has been written through Twitter messages and whole novels punched on cell phone keyboards and posted on the Web.[54]

---

49 Parrish, "Understanding Bots."

50 Ibid.

51 Ibid.

52 Ibid.

53 Marie-Laure Ryan, "Between Play and Politics: Dysfunctionality in Digital Art," *Electronic Book Review*, March 3, 2010, accessed October 23, 2015, http://www.electronicbookreview.com/thread/imagenarrative/diegetic.

54 Ibid.

She also notes another characteristic of ludic dysfunctionality that Twitter bots, particularly list-based bots, often present, "Another form of ludic dysfunctionality is the creation of a machine that performs absurd tasks."[55] Reposting all the words in a dictionary one by one is an extensive, absurd task that a human does not even dream about completing. The same applies to all the other bots posting the Bible in 140 characters at a time or every Unicode character that exists. In these cases, the absurdity comes from the method instead of the content of individual updates: the words, symbols or text snippets posted are not absurd but the process of posting them is.

Ludic dysfunction can open up new ways for technology to function, as it merely creates different kinds of functionality in tools created for practical use, "Common to all these cases is the negation of real-world practicality and the creation of a new functionality – the autotelic and self-reflexive functionality of art. Ludic dysfunctionality is only dysfunctional as far as it rejects the subordination of technology to material pursuits."[56] The inclusion of bots on our social media feeds can point out the algorithmic nature of all the other types of generated text we encounter online all the time—ads, machine-generated headlines and curated feeds, as noted by Parrish[57]—and provide a relief from the monetisable brand content present on our Twitter feeds, "In the best of cases, dysfunctionality can reach a higher functionality (for art can indeed be useful, as long as it is not in a crassly material way) by making users aware of the codes and processes (technological, linguistic, cultural and cognitive) that regulate our social and mental life."[58] The *Everyword* bot, the *Thinkpiece Bot* and even the *Magical Realism Bot* all question some aspects of social networking, media, language and creativity in their own way.

Bots as a Folk Practice

Allison Parrish notes how botmaking is a form of folk practice often completed outside of the traditional academic or artistic institutions, "[Bots] are *frequently deployed as folk practice*, outside of 'mainstream' practices of artists, academics, researchers, engineers, etc."[59] She goes on to describe how the tools that have been created to aid botmakers show the interest in the field from people with no technical background in computer science or coding, "[T]he existence and proliferation of Twitter bot-making tools like Cheap Bots Done Quick! show that there is a great deal of need and hunger for non-technical users to engage in the medium with bots."[60] There are, indeed, plenty of technologies that help non-coders make their own bots. The aforementioned *Cheap Bots Done Quick!* platform, created by George Buckenham, has been described as "a free and radically accessible botmaking tool,"[61] making it possible for any user to create a Twitter account and turn it into a bot with a few clicks. The platform uses another technology called *Tracery* to build grammar-based bots that post updates at regular intervals and are able to reply to users mentioning the bot in their posts. *Tracery* is a project by Kate Compton, a doctoral student of computer science, which provides a simple way of encoding grammars to be used for

---

55 Ibid.

56 Ibid.

57 Parrish, "Understanding Bots."

58 Ryan, "Between Play and Politics."

59 Parrish, "Understanding Bots."

60 Ibid.

61 "George Buckenham / Cheap Bots, Done Quick!" Abandon Normal Devices, Accessed February 1, 2017, http://www.andfestival.org.uk/events/george-buckenham/.

bots and other textual generators. It does not presuppose any understanding of other programming languages and the instructions for building a functioning bot grammar can be printed on a single letter page.

These tools and other online tutorials, like Parrish's own tutorial on how to turn a Google sheets database into a list-based bot, lower learning curve for people wishing to learn how to make bots to the bare minimum, opening the field to authors with the technological know-how of an average web user. With their simple few-click designs, they invite authors from different backgrounds to try out their ideas instead of limiting botmaking to programmers, technological artists and computer scientists. Furthermore, projects like *Corpora* are another example of the botmaking community aiding the creative processes of other authors by providing them with useful, categorised and curated lists to work with to make their own creations. A starting botmaker need not write down all the possible colours in a list of their own as they have open access to such a list from the get-go.

The botmaking community is more spread out than the electronic literature community, which is centred around academic institutions and the Electronic Literature Organization, as many botmakers work as independent artists or make bots as side-projects. Bots are also much more accessible than works of electronic literature as web poetry and online hypertext fiction have to be listed in directories, websites and databases and must be actively sought out by the readers. Bots are readily accessible in a channel where people are already spending their time so they do not need to be listed or publicised in directories for discovery as a simple retweet can introduce a reader to a new bot. The reading of a bot also does not require an effort similar to reading other types of electronic literature: the reading of a bot is a gradual process occurring over time and the reader is exposed to the bot's output as they scroll through their timeline.

What all the mentioned frameworks also show is an understanding of bots already evident from above: understanding the source code or the technologies running the bots is required neither for reading bots nor for creating them. Rather, writing and reading bots requires some understanding of textual mechanics and the malleability of language, the similar kind of understanding of language that can be seen in the creations of Tristan Tzara, Brion Gysin, Raymond Queneau, Allison Knowles and Christopher Strachey. While before textual artists like Knowles and Gysin needed to work with people that handled the technological aspects of their creations, the open technologies available for botmaking make experimentation of this kind open to anyone interested, with plenty of tutorials available free of charge. What a bot-authors need to concern themselves with are aspects like interestingness and variability as described above.

6. Case Studies

It is possible to condense the characteristics noted in the previous chapters into a set of elements to consider when attempting to reverse-engineer the output of bots. This set of aspects can be used as a framework to guide the analysis and comparison of generative bots. The analysis of a bot should begin with some essential questions about the nature of the bot, modified and expanded from Mark Sample's characteristisation of protest bots. When beginning to analyse a bot, one should note the information provided on the bot's profile page—the name, the Twitter account handle, the biography section and any links to additional information outside of the bot's profile. Based on the profile page, one should note the topic and the theme of the bot's output and make an educated guess about the method of textual generation. It is also worth noting whether the account makes an explicit note of the fact that it is run by a software program instead of a human author, although almost all of the most popular bots do this.

As many botmakers do not provide a source code for their bots, after making note of the initial details, the output of the bot should be analysed at large. Instead of focusing on the particularities of individual updates, a large number of updates should be compared with one another. By comparing the updates, it is possible to see the recurring words, topics and templates. This comparison can further clarify the intent of the bot, the aim it is trying to achieve. As noted earlier, there are blurry boundaries between the intents of different bots but it can be helpful to understand if the bot is presented as an artistic endeavour, as a protest or with a journalistic frame of reference. The updates should be evaluated in relation to the subject and the intent of the bot, which can be done through asking several questions of the bot's output: How is the topic reflected in the word choices and templates of the bot? How are the intent and the topic related to the chosen method of textual production and what meaning does the method add to the chosen topic? Furthermore, based on the updates and the information provided about the bot by the bot's author, the source and choice of the data in the bot should be evaluated: Are the word lists and templates hand-written by the author? Does the bot make use of an external data source and how has this source been selected? Naturally, without the full source code, not all these questions can be answered in full, but they can be used as guidelines and reminders about the nature of aspects to consider while reading the output of bots.

Naturally, the textual qualities of the updates should also be evaluated. Variability and interestingness are the main factors that make users follow bots and keep them satisfied with the scheduled outputs on their timelines. As noted earlier, grammar- and list-based bots differ greatly in that the aspects that make them interesting come from different sources and, thus, more specific questions about the artistic choices of the content produced by bots should be asked in relation to the method of textual generation. A grammar-based bot stays interesting due to the variability of updates, which is based on how extensive and non-repetitive the bot's corpora are and how varied and nested its grammar is. As noted by Chris Rodley, grammar-based bots have the ability to surprise readers with juxtapositions of concepts that a human author would not pair and this is often reflected in the corpora of grammar-based bots. Furthermore, Kate Compton notes the

key metrics by which the output of grammar-based bots can be evaluated, namely perceptual differentiation and perceptual uniqueness. As the user following a bot account does not face the updates in a long list of samples but as individual texts sprinkled amongst other posts on their timeline, perceptual differentiation is key in keeping a user following the account instead of thinking that the bot is posting the same kind of an update over and over again. As noted earlier, perceptual uniqueness, the fact that an individual generated item is memorable as such, is harder to achieve with the use of a grammar-based bot and can be more related to an individual update reaching viral popularity through retweets and sharing. At the textual level, the selection of templates and the curation of corpora should be evaluated: where do the words come from, have they been hand-picked by the author and how do they with fit the topic and the intent of the bot? The same can be asked of the boilerplate text of the grammar: in addition to how varied the sentences are, how does the choice of templates reflect the theme and aims of the bot and the attitudes of the botmaker?

As noted in chapter 4, list-based bots differ greatly from grammar-based bots in how they remain interesting to the readers. The users are often readily familiar with at least the theme of the data posted by the list-based account and the interest in the bot is based on the expectation of discovering something new from familiar data or being prompted to give one's own meaning to mundane everyday information. In the analysis of list-based bots, analysis of the data used is the prime interest. With the attitude of critical code studies, the source of the data should be assessed, be it from an open-source project or a government database. The choices made in the selection of data can clarify the intent of the bot, for example, if only a specific data point from a larger database is reflected in the bot's output. Furthermore, as noted in the case of remixes of *Everyword*-type bots, where words from extensive lists are placed into different kinds of sentences, the effect of this addition of a new context is another aspect to analyse. The determinability of the bot's output is also essential in understanding the reader's experience of following a list-based bot: A computationally-fixed bot progressing through, say, an alphabetised list has an exhaustive sensibility, performing a task which might take long but will eventually be finished. This can urge the reader to continue following an account for a long period of time. On the other hand, a randomised list lacks this determinability. The determinability of a bot's posts should also be discussed in relation to how it supports the bot's theme and intent.

The rest of this chapter is dedicated to case studies of a few exemplary bots, which will be analysed utilising the framework described above. The bots have been chosen based on their popularity and their approach to textual generation in order to explore how the choices in the design of their author-texts are reflected in their output and how the chosen methods of textual generation support the aims of the bots. Although the *Magic Realism Bot* has been discussed earlier to some extent, its actual output has been merely alluded to in the earlier sections. Thus, it will represent a bot with literary intentions in the analysis below and its output will be considered in much greater detail than before. Another bot titled *Feelings.js* will be considered as well since it also utilises the grammar-based generation method to create something novel and unexpected. Two bots by Nora Reed, *@man_products* and *@lady_products*, will be considered as examples of oppositional bots that critique the advertising industry and its use of gendered language. List- and data-based bots will be explored with two Finnish examples, my own *@kaikkionpahaa* bot and another bot, titled *@PaivanAnnos*, which both utilise the same government database as their source.

Literary Premises

The motivations and design of the *Magic Realism Bot* have been discussed in some detail already in the earlier chapters but the focus in those discussions has been on the authors' notions about the design of grammar-based bots. Thus, it is worth taking a look at the bot's actual output to see how the ideas about the specificity of grammar and the variability of templates discussed by Chris and Ali Rodley are visible in the texts generated by the bot's program. To begin with, the technologies used by the bot, the bot is written in Python with a MySQL database of grammar templates[1] and it is a perfect example of a standard, batch-mode variable text generator bot. As noted already before, the bot creates microstories or premises for narratives that fit within the 140-character limit set by Twitter.[2]

Despite its name, the bot not only publishes Borgesian magic realist stories as its scope has been expanded to include narratives inspired by other literary genres, "In other words, keeping it to Borgesian ideas (which are quite dry, rather than colourful and visual, and also metafictive) was too limiting. Currently the bot draws inspiration from a bunch of magic realist authors as well as other genres (fantasy, children's lit, YA, detective fiction)."[3] This variety of genres is easily seen in the updates of the bot: While updates like "There is a supermarket in Moscow. Instead of groceries, it sells dreams"[4] exhibit the essence of magic realist fiction by modifying an everyday scene with a supernatural detail, there are also updates that show the typical genre markers of detective fiction ("A bank manager is found murdered in an opera house. Beside her is a cannabis plant and a dead hobbit. Can you explain what happened?"[5]), riddles ("Solve this riddle: What has a knee and a kidney, but not a shoulder?"[6]) and fairytales ("An old lady is hiding in a garden. She is thinking about photographs. An elf is doing nothing at all behind her."[7]). The expansion of the bot's scope to stories very different from magic realism shows the authors' concern for keeping the followers entertained and surprised, making it one of the bot's aims to keep the followers engaged in following the bot's updates. The bot is clearly an experiment in storytelling within the affordances of the tweet format. It is also an experiment in machine storytelling in that, by choosing the grammar-based method for generating the bot's updates, the bot attempts to unleash the potential of machine-generated language to create story proposals that contain juxtapositions that a human author would be unlikely to put together, "Another reason why the bot is a good fit for generating magic realist premises is that it can create links between ideas and concepts that aren't normally connected, that are totally divergent, in a way that human don't easily do."[8]

When reading through the mass of updates generated by the bot, posted to the *@MagicRealismBot* account once every two hours, it is easy to notice that there is a very large variety of grammar templates that the bot uses. Although it is difficult to decipher the actual templates used by the bot's program, exactly identical templates appear on the bot's timeline approximately once in a hundred updates, which at the bot's current posting schedule adds up to several days between similar posts. However, although the templates and narrative proposals

---

1 Rodley and Rodley, "In Conversation."
2 Ibid.
3 Ibid.
4 https://twitter.com/MagicRealismBot/status/724342266618224640, accessed April 6, 2017.
5 https://twitter.com/MagicRealismBot/status/848206475411324928, accessed April 2, 2017.
6 https://twitter.com/MagicRealismBot/status/848146073763721216, accessed April 2, 2017.
7 https://twitter.com/MagicRealismBot/status/847934668066271232, accessed April 2, 2017.
8 Rodley and Rodley, "In Conversation."

generated by the bot are varied at large, many of them utilise similar kinds of structures to modify the concepts placed in the boilerplate text. For example, objects in the updates are often modified by subordinate clauses beginning with 'that' as an attempt to create more variety in the stories,

- "An Austrian philosophical treatise describes a way of tying a knot that wishes to be a duchess."[9]
- "A teenage king is building a circus tent that is filled with truth and beauty."[10]
- "An Albanian queen owns a chapel that is filled with your heart's desire."[11]

There is also a structure that is often used to create juxtapositions in which one object is described to be 'made of' or 'filled with' another object or an intangible concept,

- "A gifted chef bakes a cake made of butterflies."[12]
- "A hospital made of philosophy appears in Calcutta."[13]
- "A clerk falls into a pool filled with dictionaries. Nobody misses him."[14]
- "A guilt-stricken dentist falls into a swimming pool filled with human rights. She is never seen again."[15]

These kinds of repetitive modifier structures appear in the bot's output much more commonly than the full story templates, making the updates feel much less varied to the reader than they actually are. The vocabulary used also has similar problems: although the vocabulary seems to contain a large variety of distinct concepts and objects, they are often paired with modifiers that are very often either places or nationalities,

- "There is a stonemason in Burma who has a porcelain penis."[16]
- "In Mecca there is an elm tree which used to be a Norwegian politician."[17]
- "In Budapest is a schoolmaster whose heart is a piano."[18]

This kind of repetition of words of the same kind makes the reader treat nationality or place names as a staple in the updates produced by the bot, further reducing the diversity of the potential updates they reverse-engineer in their mind.

While updating of the bot to include different kinds of premises increases the variety of its output, the expansion also causes problems with the style of the produced texts. As noted by Rodley in the previous chapter, the corpora used by the bot has been hand-picked by the authors to include words that evoke a specific style and meaning.[19] It is easy to note the words related to the original magical realist storylines as many of the bot's updates contain words that bring to mind a time different from ours: for instance, people are often referred to by their vocation or title

---

9 https://twitter.com/MagicRealismBot/status/845881001649750016, accessed April 2, 2017.
10 https://twitter.com/MagicRealismBot/status/847753462032982018, accessed April 2, 2017.
11 https://twitter.com/MagicRealismBot/status/845125980104540160, accessed April 2, 2017.
12 https://twitter.com/MagicRealismBot/status/848417881972920321, accessed April 2, 2017.
13 https://twitter.com/MagicRealismBot/status/846847432222658560, accessed April 2, 2017.
14 https://twitter.com/MagicRealismBot/status/843132725405995008, accessed April 2, 2017.
15 https://twitter.com/MagicRealismBot/status/846062207297249281, accessed April 2, 2017.
16 https://twitter.com/MagicRealismBot/status/848538686207676416, accessed April 2, 2017.
17 https://twitter.com/MagicRealismBot/status/847421250574077952, accessed April 2, 2017.
18 https://twitter.com/MagicRealismBot/status/847028637031280640, accessed April 2, 2017.
19 Rodley, "Introducing Chris Rodley."

and words commonly seen in older novels and period pieces, like 'courtesan', are often used, "A witch is writing a list of people she plans to kill: A historian, a courtesan and a professor of English literature."[20] Objects that are no longer used—almanacs, dictionaries, hourglasses—also appear from time to time. The updating of the bot's templates has caused a notable shift in the vocabulary as the bot's updates today include many contemporary words, such as "HR manager,"[21] "Apple Store"[22] and "Candy Crush Saga"[23], although in an interview Rodley has mentioned how the words have been hand-picked to avoid more modern words like "Swatch watches."[24] While the different genres of stories are a welcome addition to the bot's source, one would expect the vocabulary to be more reflective of these genres rather than diluting the specificity of the bot's corpora with more contemporary concepts.

The selected method of textual generation fits with the bot's aims perfectly. The use of grammar templates with a large vocabulary enables the bot to create unexpected concepts that a human might not put together. The results of the mechanical pairing of concepts without any consideration about their meaning are, of course, varied. The bot is able to produce updates that are complex and that contain certainly unexpected pairings,

- "A 4th century BC pope walks into a forest and discovers a sandalwood tree made of social anxiety."[25]
- "A Somali maestro conducts a Donizetti opera about a tampon that lasts for 200 million years."[26]

A pope that precedes Christ is definitely a concept unlikely to be seen in human-authored fiction as is an opera about tampons. However, these kind of updates with random concepts clearly filling the blanks in a boilerplate text are so out of this world that they do not even entice the reader to start thinking about how the stories would continue. However, some of the bot's updates do present very enticing narrative premises,

- "An innkeeper spends all her spare time making a list of every sin that has taken place in Japan."[27]
- "You are transported to a city which looks like Shanghai, but everything is yellow. You can hear the faint sound of a blues song."[28]

What makes the latter updates different from the earlier ones is that they pair concepts that keep up with the theme of the narrative. It is easy to start imagining a magic realist story set in mediaeval Japan with an obsessive innkeeper. The second update verges on lyricism by accidentally juxtaposing two colour words with completely different meanings. As it can be seen, the readability and meaningfulness of the bot's updates varies greatly but this does not matter as both types of results can be expected from such experimentation with machine storytelling,

---

[20] https://twitter.com/MagicRealismBot/status/846152809837838337, accessed April 2, 2017.
[21] https://twitter.com/MagicRealismBot/status/846998436381687809, accessed April 2, 2017.
[22] https://twitter.com/MagicRealismBot/status/847632657873739776, accessed April 2, 2017.
[23] https://twitter.com/MagicRealismBot/status/845276983747489793, accessed April 2, 2017.
[24] Rodley, "Introducing Chris Rodley."
[25] https://twitter.com/MagicRealismBot/status/847270244888399873, accessed April 2, 2017.
[26] https://twitter.com/MagicRealismBot/status/848115872505839616, accessed April 2, 2017.
[27] https://twitter.com/MagicRealismBot/status/846575624168755200, accessed April 2, 2017.
[28] https://twitter.com/MagicRealismBot/status/846213211523203073, accessed April 2, 2017.

especially when considering the mechanical way the bot combines the concepts in the first place, with little regard to the meaning of the concepts other than the kind of word lists they have been included in.

On the whole, the *Magic Realism Bot* is a varied and complex experiment in generative storytelling whose output shows both the successful and unsuccessful creations with equal weight. However, although the templates and vocabulary used in the bot's program are varied on their own, the bot's output shows that the complexity and variability of the modifier phrases shared between many different top-level templates affect the reader's experience of the text-to-be-seen to a great extent. In the case of this bot, the attempt to add complexity with subjunctive clauses and modifier adjectives actually reduces the perceptual differentiation between updates due to the overuse of specific words and formulas that stick to the reader's mind. Due to the repetitive phrasings, the bot does not appear as interesting and multifaceted in the long run as the complexity of its program would allow, making it risk its readers unfollowing the account sooner than they would if the modifying phrases were either used less often or expanded to include a much more varied set of words.

All the Feelings

> today, i feel like a shareable submission[29]

Another bot that displays behaviour similar to the *Magic Realism Bot*'s randomised juxtapositions of concepts is Katie Rose Pipkin's early *Feelings.js* bot (*@feelings_js*). In a way, it could be treated as another example of experimentation in mechanical storytelling as the bot uses a large variety of words about concrete objects and abstract concepts to describe new emotions and feelings. The bot's lowercase output with strange combinations of adjectives and objects resembles odd internet speak that one might encounter on Tumblr. Although the bot's bio, "a little bot that feels many odd things every day,"[30] points out the bot as the one experiencing the feelings, the format of the output with the use of first-person narration also makes the reader relate to the content, identify with it and share the most successful posts to their own followers with a retweet. Furthermore, the first-person phrasing used in the updates enables the bot's output to blend in with the intention-typical posts that appear on a reader's timeline since it is very possible for a fellow human to start their posts with the words 'today, i feel like a...'

The bot's output is very uniform in style, repetitively formulaic with a few different options of sentence templates that vary in length and amount of additional details. Each update contains one concept or concrete thing that serves as the description of the feeling. This main word can be modified by an adjective that precedes the word or a prepositional phrase with another thing or concept that follows it. An update can also include both of the modifiers at the same time. The algorithm of the bot also chooses the beginning statement from a range of options which vary the verbs, the use of double negatives and the placement of commas, for example. This variability in the bot's grammar creates updates that can range from the shortest "a spectacle"[31] and "today a

---

[29] https://twitter.com/feelings_js/status/848508323024523265, accessed April 2, 2017.

[30] https://twitter.com/feelings_js, accessed April 6, 2017.

[31] https://twitter.com/feelings_js/status/843072497763409920, accessed April 2, 2017.

transmission"[32] to one of the longest possible updates "at the moment i feel not unlike a faint-hearted committee during a commonwealth."[33] Most of the updates are, however, in between these extremes, possibly starting with an interjection or a conjunction and continuing with the words 'i feel' or 'i'm feeling',

- "well i'm feeling close to an earthlike arrow inside a handball"[34]
- "so, i feel like a remuneration"[35]
- "today i feel like a hypnosis"[36]
- "i'm feeling close to a contrivance"[37]

The format of the updates is clearly recognisable on one's timeline and yet the subtle changes in comma placement and phrasing make the output vary enough not to be monotonous and much too mechanical.

It can be noted from the lack of diversity in the boilerplate text that the focus of the bot is on the corpora instead. The corpora used by the bot is extremely varied: in the over 200 examples examined in detail for this analysis, only a couple of main words like "juniper"[38] and "impotence"[39] repeat in two updates. Variety of word choice to this extent with a grammar-based bot suggests either the use of very large corpora, a database or an API[40] that links to a large dictionary. This is confirmed by Pipkin in an interview about the bot, "Feelings.js (and a few others like it) is basically a fill-in-the-blank Wordnik wrapper. It has a variety of possible sentence structures on a switch statement, and then pulls parts-of-speech from the dictionary API."[41] As noted before, Wordnik is a large dictionary database with which programmers can link their bot programs in order to gain access to searchable, categorised word lists.

The bot's program seems to choose the words randomly without much determinability, as the updates do not progress through word lists alphabetically or based on a specific topic. Due to this random selection, the reader does not get a sense of the scale of the source material and is unable to see if the themes of the selected words have been limited in some way. The words used range from very concrete things like "a navel"[42] and "a muslin"[43] to "sincerity"[44] and "a lengthy

---

[32] https://twitter.com/feelings_js/status/847451353651617792, accessed April 2, 2017.
[33] https://twitter.com/feelings_js/status/844371060753666049, accessed April 2, 2017.
[34] https://twitter.com/feelings_js/status/847934536356954112, accessed April 2, 2017.
[35] https://twitter.com/feelings_js/status/847602347446476800, accessed April 2, 2017.
[36] https://twitter.com/feelings_js/status/847300363623383042, accessed April 2, 2017.
[37] https://twitter.com/feelings_js/status/843918077855252480, accessed April 2, 2017.
[38] https://twitter.com/feelings_js/status/848387520178122752 and https://twitter.com/feelings_js/status/847572151020404736, accessed April 2, 2017.
[39] https://twitter.com/feelings_js/status/845126031790895104 and https://twitter.com/feelings_js/status/842800708835561472, accessed April 2, 2017.
[40] API stands for Application Programming Interface, which, in this case, means a method of connecting a software program to an external system that can provide it with data and run algorithms with more processing power than the program itself.
[41] Katie Rose Pipkin, "About a Bot: Interview with Katie Rose Pipkin," interview by Taina Bucher, *Furtherfield*, December 7, 2015, accessed March 30, 2017, http://furtherfield.org/features/interviews/about-bot-interview-katie-rose-pipkin.
[42] https://twitter.com/feelings_js/status/848176129223655425, accessed April 2, 2017.
[43] https://twitter.com/feelings_js/status/845458221657722887, accessed April 2, 2017.
[44] https://twitter.com/feelings_js/status/845488418138308608, accessed April 2, 2017.

elocution beneath a liquidity."[45] Also, the logic behind the pairing of the main word and the modifying adjective is fuzzy as some updates utilise adjectives that fit with the main word,

- "at the moment i feel like a salty kisser"[46]
- "well i'm feeling close to a foursquare server"[47]
- "i'm feeling close to a nonperishable vinaigrette"[48]

while other updates pair the words with no clear deliberation about their meaning,

- "i think i feel not unlike an established salad"[49]
- "so i'm feeling sort of like an australopithecine parlor"[50]
- "today a pleasure-boat dissipation with a waxwork"[51]

Although it is certainly possible that there is a function placing meaningful pairs of adjectives and objects between randomly selected ones, the ratio of meaningful pairs to random ones does suggest that the placement of adjectives is random and the meaningful updates are infrequent accidents. Furthermore, even many of the more sensible pairs depend on the reader to make them meaningful as they are certainly not commonly used in everyday, intention-typical language.

The use of a large dictionary in a grammar that gives minimum variability in sentence structures and theme makes the *Feelings.js* bot resemble the *Everyword* bot. The random use of a dictionary and the mechanical pairing of adjectives and concepts gives the bot's readers an opportunity to give new meanings to the combinations offered by the bot. The discovery of juxtapositions that are meaningful to the reader plays a great part in reading the bot's output. Furthermore, by focusing on the theme of feelings, the random pairs of words gain a new context, which makes the use of technical, field-specific or mundane words very poetic. Like with *Everyword*, reading the bot's output is not centred merely on finding new concepts one can relate to but it is also highly related to sharing the interesting, entertaining and successful updates to one's own followers. Especially the shorter updates generated by the bot are readily shareable as descriptions of the reader's own mental landscape,

- "today i feel kind of like an inferno"[52]
- "today i feel not unlike an energy"[53]
- "now, a booze"[54]

The output of *Feelings.js* emphasises the mechanical nature of its text production at every turn: the lack of variety in templates, the superhumanly large selection of concepts and their random pairing all point out to the fact that the odd feelings are generated by a machine. This emphasis

---

[45] https://twitter.com/feelings_js/status/842649714218729473, accessed April 2, 2017.
[46] https://twitter.com/feelings_js/status/845397823067688963, accessed April 2, 2017.
[47] https://twitter.com/feelings_js/status/847723162460504064, accessed April 2, 2017.
[48] https://twitter.com/feelings_js/status/846726589643673601, accessed April 2, 2017.
[49] https://twitter.com/feelings_js/status/845065635503783936, accessed April 2, 2017.
[50] https://twitter.com/feelings_js/status/847119163328024576, accessed April 2, 2017.
[51] https://twitter.com/feelings_js/status/844461652464582658, accessed April 2, 2017.
[52] https://twitter.com/feelings_js/status/844431482680758272, accessed April 2, 2017.
[53] https://twitter.com/feelings_js/status/844401256684797952, accessed April 2, 2017.
[54] https://twitter.com/feelings_js/status/843344289526636544, accessed April 2, 2017.

on the mechanical nature of the language reduces the audience's expectation of coherence: the updates are allowed to be confusingly long or contextually unfitting as long as some of the new concepts produced by the bot's algorithm let the readers discover something new and poetic about themselves or the world around them.

Ridiculing the Advertising Lingo

Nora Reed is a prolific bot author and their oppositional *Thinkpiece Bot* has been much discussed in the earlier chapters. Many of their other bots are also oppositionally-motivated and they utilise the grammar-based method to amplify the message of the bot. To provide an example of analysing oppositional bots, this analysis will focus on comparing two similarly-themed bots by Reed, namely *Man Products* (*@man_products*) and *Lady Products* (*@lady_products*).[55] As suggested by the names of the bots, they target the advertising industry and its use of language in reaching out to gender-based segments of consumers. Both of the bots attempt to ridicule both gendered marketing in general as well as the formulaic language used in it through the affordances of the grammar-based method: the bots make much use of the potential for the absurd juxtaposition of disparate concepts and the formulaic nature of boilerplate texts.

Although both of Reed's bots embody the same theme and objectives as well as some basic word lists, they differ greatly in the complexity of the source, which can be seen from the variety of outputs produced by the two bots. Firstly, *@man_products*, the updates of which are posted in all capital letters, makes use of a very wide array of templates from which the bot generates its advertising slogans. Some of the updates contain very traditional advertising language, like "THE STRONG FEELING OF WHEY AND BERRIES, TOGETHER AT LAST,"[56] and some make their point by attaching masculine-sounding attributes to products, like "UNWASHED LACTAID [...],"[57] "ENRICHING-FLAVORED STEROIDS"[58] and "LEATHER-FLAVORED CUSHIONS."[59] Some other formulas used by the bot take the stance of allowing men to use some product previously unavailable to them,

- "WITH MANLY LOOK DUDES CAN FINALLY BUY BEER"[60]
- "WITH SUPERCHARGED HOME GUYS CAN FINALLY CONSUME COLD COMPRESS [...]"[61]

With the random placement of different goods in this formula, the absurdity of the advertising style starts to show, as in the example above, where beer, a traditionally male-associated product, is marketed as a novelty for men. Another type of this same formulation is taking a traditionally

---

[55] There is also a third bot that produces marketing language, titled *Luxury Products* (*@luxe_products*), but it will not be discussed here for the lack of space and for its similarity with the other two bots.

[56] https://twitter.com/man_products/status/845388323900342274, accessed April 10, 2017.

[57] https://twitter.com/man_products/status/848242131315691520, accessed April 10, 2017.

[58] https://twitter.com/man_products/status/843168715197874177, accessed April 10, 2017.

[59] https://twitter.com/man_products/status/843214016881328128, accessed April 10, 2017.

[60] https://twitter.com/man_products/status/851231828480581632, accessed April 10, 2017.

[61] https://twitter.com/man_products/status/850597651376721920, accessed April 10, 2017.

feminine product and modifying that with a stale staple phrase—a technical detail, camouflage pattern or bacon flavour, for example—commonly used in marketing targeted to men,

- "HIGH FIDELITY MAXI SKIRT: FINALLY AVAILABLE FOR GUYS [...]"[62]
- "BACON-ENCRUSTED PERSONAL MOISTURIZER: FINALLY AVAILABLE FOR GUYS [...]"[63]
- "FINALLY, PRENATAL VITAMINS FOR GUYS! NOW IN POWER!"[64]
- "CAMO MASCARA: FINALLY AVAILABLE FOR DUDES"[65]

Straightforward advertising slogans are also produced by the bot's algorithm as phrases like "[...] PUMP THE HUT"[66] and "DOMINATE YOUR HOUSE"[67] are placed in some of the updates.

Whereas *@man_products* utilises a large grammar with a wide variety of potential formulas, *@lady_products* is much more limited with its sentence types. Although the form of the templates might differ slightly between updates, the understated lowercase output of the bot mainly produces randomised advertising lingo in varying lengths,

- "finally. makeup"[68]
- "bathe in maxi skirt"[69]
- "pluck your with CoQ10"[70]
- "immerse yourself in pink with fuchsia facials"[71]
- "Dead Sea inspired magic. with the essence of the future"[72]
- "coral reef inspired foundation. now with all-natural green"[73]

The only very notable template used in the bot is the 'now available in pink/black' formula, which makes fun of the way products are often customised for women by making a version of them in pink. *@lady_products* uses this formula with more interesting, often intangible concepts,

- "now available in pink: friends! pink butches"[74]
- "now available in pink: chemistry! Unisex necromancy"[75]
- "now available in black: the void! shimmery tiaras"[76]
- "cherry-scented rainbows. now available in black: rainbows"[77]

---

[62] https://twitter.com/man_products/status/851322427875479553, accessed April 10, 2017.
[63] https://twitter.com/man_products/status/850461756971507715, accessed April 10, 2017.
[64] https://twitter.com/man_products/status/848921608421081088, accessed April 10, 2017.
[65] https://twitter.com/man_products/status/849555786481139712, accessed April 10, 2017.
[66] https://twitter.com/man_products/status/848015630443962368, accessed April 10, 2017.
[67] https://twitter.com/man_products/status/850099367512035328, accessed April 10, 2017.
[68] https://twitter.com/lady_products/status/848060953107333120, accessed April 10, 2017.
[69] https://twitter.com/lady_products/status/850914765732007937, accessed April 10, 2017.
[70] https://twitter.com/lady_products/status/850733570494787584, accessed April 10, 2017.
[71] https://twitter.com/lady_products/status/850688271122395136, accessed April 10, 2017.
[72] https://twitter.com/lady_products/status/845614841234640897, accessed April 10, 2017.
[73] https://twitter.com/lady_products/status/849102827326668800, accessed April 10, 2017.
[74] https://twitter.com/lady_products/status/847562676058300420, accessed April 10, 2017.
[75] https://twitter.com/lady_products/status/848785735826198528, accessed April 10, 2017.
[76] https://twitter.com/lady_products/status/850597676550955009, accessed April 10, 2017.
[77] https://twitter.com/lady_products/status/843259338617688065, accessed April 10, 2017.

The lack of diversity in the different types of updates generated by the *@lady_products* results in added importance in its word choices. In general, when facing a similar lack of variety in templates, the analysis of the bot should pay special attention to the range of language used in the corpora of the bot. With *@lady_products* a large variety of language would be essential as the comical juxtapositions are not created at the grammar-level, like with the bacon-flavoured AK-47s of the *@man_products* account, but at the word-level.

Not only is *Man Products* very varied in its sentence templates, also the corpora used in the bot comprise an exquisite range of items and concepts. Many updates make use of the absurdity of adding male-centric modifiers to completely conventional goods, like beer or Lactaid, and female-associated items. These modifiers range from leather, metal and other 'masculine' materials to the ninja imagery and technical efficacy of the products,

- "CRUSH HIS OXYGENATION WITH NINJA CANDLES"[78]
- "WOW: XTREME-FLAVORED JUICE [...]"[79]
- "DAMN: FIREMAN-CERTIFIED-FLAVORED CANDY"[80]

The bot also attempts to expose the violence in the language used in male-targeted advertising by including plenty of violent verbs in the slogans produced by the bot. This violent language is translated into ridiculous statements by the bot's sentence templates,

- "[...] CRUSH THE DOG"[81]
- "[...] MAIM YOUR ARSENAL"[82]
- "ASSAULT YOUR WORLD WITH NO HOMO PENILE ENHANCEMENT"[83]

In addition to emphasising the violence in advertising language, the vocabulary is also used to attack so-called 'bro culture' at large. Words like "dude" and "bro" are repeated to the point of meaninglessness and terms like "libertarian"[84] and "misogynist"[85] are casually included to describe the products.

The corpora of the bot also include refreshing and surprising references to literature and art, which are used in a lot of different kinds of templates,

- "WITH FIREMAN-CERTIFIED SELF-LOATHING BROS CAN FINALLY BUY FICTION"[86]
- "RUSH HER BLOOD WITH EXTRA-CONCENTRATED FICTION"[87]
- "LIFEHACK HIS OXYGENATION WITH UNYIELDING ART"[88]

---

[78] https://twitter.com/man_products/status/848423325877641216, accessed April 10, 2017.
[79] https://twitter.com/man_products/status/848106224780529664, accessed April 10, 2017.
[80] https://twitter.com/man_products/status/848287430377701376, accessed April 10, 2017.
[81] https://twitter.com/man_products/status/849329291636682752, accessed April 10, 2017.
[82] https://twitter.com/man_products/status/849102802152501249, accessed April 10, 2017.
[83] https://twitter.com/man_products/status/846792578662043650, accessed April 10, 2017.
[84] https://twitter.com/man_products/status/846475489988530176, accessed April 10, 2017.
[85] https://twitter.com/man_products/status/845478921571356672, accessed April 10, 2017.
[86] https://twitter.com/man_products/status/845071235876864004, accessed April 10, 2017.
[87] https://twitter.com/man_products/status/847698547075985408, accessed April 10, 2017.
[88] https://twitter.com/man_products/status/851005338249768960, accessed April 10, 2017.

- "SMOKE-FLAVORED TELEVISION [...] SATISFY THE NARRATIVE"[89]

The addition of these terms takes the bot away from the field of general advertising and helps to create more ridiculous juxtapositions as literature is hardly even marketed to men, let alone with such efficacy-accentuating properties.

As noted earlier, because of the lack of variety in the templates of *@lady_products*, the vocabulary used in the bot is bound to gain more detailed attention from the reader. However, in this closer analysis, the vocabulary of the bot presents a fairly narrow selection of words and themes. Much of the updates use words that one would see in real world advertising: cosmetic products, clothes and accessories are present and they are modified by common advertising buzzwords focusing on the artisanal, refreshing or pampering qualities of the product,

- "home roasted juice: it's time to refresh yourself"[90]
- "every day, flutter your eyelashes with citrus cushions"[91]
- "love sunglasses: don't forget to primrose"[92]
- "shimmering rum. cleanse with nature"[93]

Although the combinations of words allowed by the bot's source can be unexpected—citrus cushions and rum cleanse are not concepts one often sees in advertisements—they do not push the absurdity of the marketing slogans to the same level as the combinations of the *Man Products* bot. Rather, these kinds of updates merely seem to repeat the advertising buzzwords they are aimed to criticise.

The vocabulary of the *Lady Products* bot does offer some surprises in its themes. There are common references to the field of magic, witchcraft and the occult in many of the updates,

- "pluck your witchcraft"[94]
- "sweet necromancy. it's time for bats"[95]
- "shameless blood offerings: fuck blush"[96]
- "now with all-natural flowers! dermatologist recommended blood offerings"[97]

Even though the inclusion of blood offerings in language more commonly focused on self-care tips can result in absurdity, the list of words referring to the occult is so short that the aforementioned concepts repeat in the updates with very short intervals. The frequency of use reduces the words' potential for surprise in the long run. Furthermore, as the words related to magic and witchcraft appear in exactly the same formulas that produce the straightforward advertising slogans noted above, the placement and use of this specific field of language does not look very deliberate, reducing the absurdity of the inclusion of witchcraft and necromancy in the bot's selection of language even more.

---

89 https://twitter.com/man_products/status/847245565511426050, accessed April 10, 2017.
90 https://twitter.com/lady_products/status/849691706069049344, accessed April 10, 2017.
91 https://twitter.com/lady_products/status/848921633498824704, accessed April 10, 2017.
92 https://twitter.com/lady_products/status/847879759409360897, accessed April 10, 2017.
93 https://twitter.com/lady_products/status/847517379735273474, accessed April 10, 2017.
94 https://twitter.com/lady_products/status/851005363398856705, accessed April 10, 2017.
95 https://twitter.com/lady_products/status/850235290123771904, accessed April 10, 2017.
96 https://twitter.com/lady_products/status/850280587348779008, accessed April 10, 2017.
97 https://twitter.com/lady_products/status/850642975969423360, accessed April 10, 2017.

There are also some references to body politics included in the bot's output. Many of the outputs use "sapphic"[98] as an attribute of the products while butches and heteronormativity appear in updates like "every day, heteronormativity with non-stick butches."[99] These references are also outperformed by the straightforward advertising language, just like the references to the occult. All in all, the bot's output contains a much larger number of updates that repeat real-world cosmetics advertising lingo than slogans that contain unexpected concepts and absurd juxtapositions. When compared with *@man_products*, whose political nature is reflected throughout the bot's grammar and corpora, *@lady_products* does not push the oppositional absurdity nearly to the same level.

In conclusion, the *Man Products* bot makes references to a much wider variety of concepts than the *@lady_products* account. The slogans produced by the bot push even the violence and aggression present in a lot of advertisements to men to absurdity, with updates like "[...] OWN THE ENEMY WITH RUGGED JEWELS [...]"[100] and "FIREMAN-CERTIFIED MEAT [...] PULVERIZE YOUR TESTOSTERONE."[101] Although the output contains a wide variety of different types of sentences, they are all recognisable as being inspired by advertising language. The grammar-based generation of these slogans is a perfect fit to achieve the goal of criticising this type of language as the machine-generated output reminds the reader of the formulaic nature of advertising slogans. Furthermore, the highly-curated corpora of the bot also help to enhance this notion of formulaic advertising by using a well-recognisable style to introduce highly unexpected 'man products'— like "SWEAT INDUCING BODY CREAM: FINALLY AVAILABLE FOR MEN [...]"[102] and "CROSSFIT BUTTPLUGS [...] ELECTRIFY YOUR BOD"[103]—to the reader.

Lady Products introduces the following kinds of products to the reader,

- "powder yourself with bath bombs"[104]
- "fuck pink with fair trade body powder"[105]
- "luxuriate in blush with sapphic cease and desist orders"[106]

While "sapphic cease and desist orders" are certainly unexpected, much of the bot's output is unvaried and merely repeats the vacuous advertising lingo it is aimed to criticise. The unvaried grammar makes the reader focus on the aspect that does change from one update to another, namely the corpora, but the vocabulary used in the bot lacks the political flavours and the sarcasm of the *Man Products* bot. While the updates of *@man_products* are filled with well-recognised tropes of male-targeted advertisements, their counterparts are surprisingly lacking from the *@lady_products* bot's output: merely the obsession with presenting everything in pink versions is present, although female-targeted advertising includes many other recognisable tropes, like, for example, the equation of depilation with so-called 'me time.' Not only is the *@man_products* bot

---

98 https://twitter.com/lady_products/status/842851650792374272, accessed April 10, 2017.
99 https://twitter.com/lady_products/status/848559243984859137, accessed April 10, 2017.
100 https://twitter.com/man_products/status/850642950791077888, accessed April 10, 2017.
101 https://twitter.com/man_products/status/849918175965831168, accessed April 10, 2017.
102 https://twitter.com/man_products/status/844527652673327104, accessed April 10, 2017.
103 https://twitter.com/man_products/status/846158401386242048, accessed April 10, 2017.
104 https://twitter.com/lady_products/status/850144691966402561, accessed April 10, 2017.
105 https://twitter.com/lady_products/status/850189989111513088, accessed April 10, 2017.
106 https://twitter.com/lady_products/status/848015655647490048, accessed April 10, 2017.

more varied in its grammar and vocabulary, it also uses the potential for absurdity allowed by the grammar-based method to its success, whereas *@lady_products* misses the mark in all the analysed aspects.

Finding the Fun in Government Data

All the bots above have been grammar-based so the final two bots analysed are list- or data-based. To exemplify the differences that can be found in evaluating list-based bots, a comparison will be made between two bots that utilise the same source data, which is a database of nutritional values of food items maintained by the National Institute for Health and Welfare in Finland. The database contains the names and nutritional values of thousands of foods and is available for any use under a Creative Commons CC-BY licence. Both of the analysed bots explicitly cite this data source in the biography section of the account.

The first bot, my own *Kaikki on pahaa*[107] account (*@kaikkionpahaa*) follows a simple formula readily evident in the name of the account: the now-defunct account posted the names of 1,085 food items followed by the words 'is bad,' making the account similar to the remixes of the *Everyword* bot. The updates of the bot are posted in an alphabetical order, making the bot computationally-fixed and exhaustive in nature. A sampling of the bot's output includes updates such as,

- "Dark green leafy vegetable is bad."[108]
- "Date is bad."[109]
- "Tomato sauce is bad."[110]

Whereas *Kaikki on pahaa* account's biography section contains no other information than credits to the data source and the author, the *Päivän annos*[111] account (*@PaivanAnnos*) is presented with the premise "What one should eat in a day to get everything one needs."[112] followed by the note "[T]he results are unscientific."[113] The *@PaivanAnnos* account also utilises a template that remains the same in all of the updates. However, rather than being merely list-based, the bot uses the nutritional values included in the database to calculate how much of a randomly-selected food one should eat to consume the recommended daily amount of some vital nutrient. Although the bot does not make this clear, it can be assumed that the calculations are also done according to some government-sanctioned nutritional guidelines. The results of these calculations are presented in updates such as,

---

[107] "Everything is bad."

[108] "Tummanvihreä lehtikasvis on pahaa." https://twitter.com/kaikkionpahaa/status/810954999211708416, accessed April 2, 2017.

[109] "Taateli on pahaa." https://twitter.com/kaikkionpahaa/status/810351020152864768, accessed April 2, 2017.

[110] "Tomaattikastike on pahaa." https://twitter.com/kaikkionpahaa/status/810788902776766464, accessed April 2, 2017.

[111] The name has a double meaning as it can be read both as "Recommended Daily Intake" in the context of nutritional science and as "the dish of the day" in the context of cooking and restaurants.

[112] "Mitä pitäisi syödä päivässä jotta saisit kaiken tarvittavan." https://twitter.com/PaivanAnnos, accessed April 2, 2017.

[113] "[L]opputulos epätieteellinen." https://twitter.com/PaivanAnnos, accessed April 2, 2017.

- "You get the daily recommended intake of Protein from eating 402g of Beef, with fat, average."[114]
- "You get the daily recommended intake of Calcium by eating 4,1kg of Pasta, spaghetti, naturally gluten-free, with egg."[115]
- "You get the daily recommended intake of Vitamin B6 by eating 16kg of Margarine 80%, flora for baking and cooking."[116]

Part of the interestingness of both of the bots comes from the choice of data source: although everyone is familiar with food items, the list of items contains plenty of highly specific and strange-sounding items, such as foods used as industrial ingredients, like "soy protein concentrate"[117] and "industrial deep frying fat."[118] Furthermore, there are also plenty of items one would initially not expect to be included, such as several different cocktails, like in the update "Screwdriver is bad."[119] Both of these types allow the readers to discover something new or unexpected from familiar source data. The templates used to produce the updates, in addition to the potential discovery of unexpected foods and strange phrasings, invites the followers of the account to share the updates to their own followers, especially if they disagree with the content of the update.

When comparing the two bots, it becomes evident that the *Kaikki on pahaa* bot only uses the beginning of the database's food item labelling, leaving out the details specifying ingredients, fat content and brand, which are all included in the posts by the *@PaivanAnnos* bot. This truncation of the names makes some of the items sound absurdly non-specific, like in the updates "Crumble is bad."[120] and "Light-coloured root vegetable is bad."[121] *@PaivanAnnos* plays with absurdity in a different way: Firstly, by including the full database name of the food items, the reader is forced to notice the absurd way mundane food items are classified and presented in the context of scientific institutions, with plenty of commas in between, "You get the daily recommended intake of Salt by eating 1,6kg Egg, fried, no fat, no salt."[122] Secondly, the results of the calculations often result in the bot suggesting the consumption of absurd amounts of food in order to gain the necessary amount of some obscure nutrient, like in the suggestion that the daily

---

[114] "Päivän annoksen Proteiinia saat syömällä 402g Naudanliha, rasvainen, keskiarvo." https://twitter.com/PaivanAnnos/status/846423329498157056, accessed April 2, 2017.

[115] Päivän annoksen Kalsiumia saat syömällä 4,1kg Pasta, spagetti, luontaisesti gluteeniton, kananmunaa. https://twitter.com/PaivanAnnos/status/842044476692545541, accessed April 2, 2017.

[116] "Päivän annoksen B6-vitamiinia saat syömällä 16kg Margariini 80%, flora leivontaan ja ruoanlaittoon." https://twitter.com/PaivanAnnos/status/839824850571177984, accessed April 2, 2017.

[117] "Soijaproteiini konsentraatti on pahaa." https://twitter.com/kaikkionpahaa/status/809928232116035584, accessed April 2, 2017.

[118] "Teollisuusuppopaistorasva on pahaa." https://twitter.com/kaikkionpahaa/status/810532213452378112, accessed April 2, 2017.

[119] "Screwdriver on pahaa." https://twitter.com/kaikkionpahaa/status/808810870922969088, accessed April 2, 2017.

[120] "Rouhe on pahaa." https://twitter.com/kaikkionpahaa/status/808221991505448960, accessed April 2, 2017.

[121] "Vaalea juurikasvis on pahaa." https://twitter.com/kaikkionpahaa/status/811347586779058176, accessed April 2, 2017.

[122] "Päivän annoksen Suolaa saat syömällä 1,6kg Kananmuna, paistettu, ei rasvaa, suolaton." https://twitter.com/PaivanAnnos/status/847238699804798976, accessed April 2, 2017.

need for carbohydrates can be fulfilled by consuming roughly 12 kilos of non-butter table spread.[123] Furthermore, the double meaning of the name of the bot is also meaningful in relation to the bot's content; if anyone were to follow the bot's suggestions, 12 kilos of table spread might actually end up being their dish of the day.

Although both of the bots utilise the same source data, they process the data in very different ways. The *Kaikki on pahaa* bot presents more list-like behaviour whereas *Päivän annos* also runs calculations with the data. Even though the data used is very official and from a governmental source, both of the bots emphasise the absurd aspects of the database. *@kaikkionpahaa* relies on the reader's discovery of unfamiliar items and the absurd truncation of the database labels while encouraging sharing by utilising a sentence template that readers will want to engage with. The interestingness of *@PaivanAnnos* comes from a slightly different source. There is much more variability in the bot's outputs as the items are selected at random and also the name of the nutrient included in the calculations changes. As opposed to *Kaikki on pahaa*, the latter bot not only relies on the absurdity and discovery of the items included in the database but also encourages sharing by producing absurd calculations that user will want to share.

In contrast to a lot of bots that are data-based and take a journalistic stance towards their data, what is common in both of these two bots is that even though their source is very official and dry, they manage to turn the dryness of the database into a source of entertainment that the reader wants to follow and share further to their own followers.

What Should Be Valued

All the bots analysed above work in some way with formulaic language, be it literary narrative, advertising lingo, governmental guidelines or just repetitive structures. Bots excel in this kind of repetitive, mechanical text production and provide their readers with surprises. Botmakers like Chris Rodley, Katie Rose Pipkin and Nora Reed utilise the potential for these unexpected juxtapositions to achieve a variety of aims. Bots can be used to innovate descriptions of stories, things and feelings that would never appear in texts authored by humans. Much in the same way, the unexpected juxtapositions of concepts can be used to expose and critique the vacuousness and formulaic nature of the language used in media, advertising and politics. While individual updates from the aforementioned bots can be intriguing and compelling on their own, only an analysis of the bots' updates at large can give an insight to the authorial choices made in their creation.

The framework described above can remind the reader to consider the different aspects that affect the reading experience. Like with the examples above, this kind of methodological examination of the elements of grammar- and list-based bots can show the reader how the complexity of grammar or the scope of the vocabulary are directly related to the interestingness of the generated texts. Not only does this analysis, or reverse-engineering, of a bot's output tell us about the variability of templates and the range of language used in the bot's source, it also enables the reader to further understand *how* the botmaker attempts to achieve their goals. The methods, techniques and artistic choices should always be evaluated in relation to how they either aid or hinder understanding the bot's message and achieving the bot's goals.

[123] https://twitter.com/PaivanAnnos/status/839462461573050368, accessed April 2, 2017.

All of the bots above can be treated as experiments in mechanical storytelling, harnessing the potential of random placement of concepts and the power of progressing through extraordinary amounts of data to create texts some of which would not have been considered possible by a human writer. It is natural with this kind of experimentation that not all the generated updates are successful. They can be too complex, too unrelated or plainly impossible to understand. However, varied results are what the reader of a bot grows to understand and missing the mark in some updates becomes expected. What the framework above can give to the reading of bots is a bigger picture of the aspects that the botmaker can control directly, the author-text. Instead of merely applauding occasional hits and ignoring the less desirable generations, the framework lets us appreciate the craft of botmaking and the myriad of skilled choices that go into making a bot that makes the most with repetitive, randomised and mechanical word placement.

7. Conclusions

This thesis has proposed different characteristics and aspects to be considered when analysing bots from the viewpoint of electronic literature. As there is not much theoretical research about Twitter bots, the method of looking at theories on procedurally generated texts and some historical examples of textual generators has been successful in elucidating the different processes at work in creating and running Twitter bots. Considering the seminal theories about the nature of e-lit works by Philippe Bootz, Espen J. Aarseth and Noah Wardrip-Fruin, among others, it can be noted that the process of creating bots is very similar to creating other types of textual generators. What is evident in these theories is that instead of focusing solely on the output of the generator, the text-to-be-seen, the whole design process of the generator should be evaluated.

The presentation of earlier examples of procedural text generation enables viewing Twitter bots as experiments in textual mechanics and procedural storytelling. The analyses of exemplary Twitter bots towards the end of the thesis show how reading bots is similar to reading any earlier example of textual generation in that the reader seeks to make sense of the procedures generating the text-to-be-seen to better understand how the mechanics of procedural generation relate to the topic, theme and intent of the bot. In addition to the theories and earlier examples of electronic literature, a number of notes from current botmakers and programmers have supplemented the arguments in this thesis. The interviews, FAQs and other writings by authors like Chris and Ali Rodley, Nora Reed and Kate Compton have been essential in understanding the practical considerations of crafting Twitter bots. Without the notions of these creators, the framework for analysis presented in the previous chapter would be seriously lacking in detailed understanding of the botmaker's craft.

* * *

The source code of Twitter bots is most often inaccessible to the reader as it runs on a server separate from the social network. Instead of being able to use the code inspector of a web browser like with online poetry and other web works, understanding the processes that create the bot's text-to-be-seen requires reverse-engineering its output. Allison Parrish defines this process of reverse-engineering as understanding the way the bot works. To help understanding the processes at work behind a bot's updates, this thesis has presented a number of essential characteristics of different types of bots to be considered in the analysis. The topic, the sources of the bot's data and the intent of the bot should always be considered in addition to how well the choice of the generative method complements the intent of the bot. As the method by which a bot's output is generated affects the questions that should be raised about the bot's source and data, bots have been divided into different types, all of which have their own metrics of interestingness and success. The analyses in this thesis have focused on grammar- and list-based bots, although textual generation with the help of Markov chains was also explained briefly.

Grammar-based bots are based on the random placement of words into boilerplate text described in the bot's grammar. This method is perfectly suited for experimental, mechanical

storytelling as it enables novel ideas to be automatically generated through the random combination of disparate concepts. Many of the bots discussed in this thesis utilise the grammar-based method in an artistic manner, generating narrative proposals, poetry or entertaining anecdotes. However, the grammar-based method is also well-suited for protest and societal critique. With their relentless, around-the-clock generation of new updates, these protest bots are able to oppose the vacuous, formulaic languages of advertising and media while other oppositional bots make visible the nonstop gathering of private data by government intelligence agencies. The creation of grammar-based bots is focused on staying relevant and interesting over a long period of time. Thus, botmakers spend a lot of time crafting complex, nested grammars and curating expansive corpora in order to create a large range of updates that still manage to embody the style of the bot and deliver the message through individual 140-character micronarratives that appear on the reader's timeline.

List-based bots, on the other hand, highlight the data they are tasked to post in instalments over a long period of time. These bots vary greatly in theme, iterating over thousands of individual items and allowing their readers to discover something new from familiar data. The reading of a list-based bot is distributed reading, allowing the reader to complete a superhumanly long task with the aid of a machine. The analysis of a list-based bot focuses mainly on the curation work performed by the bot's author, noting the sources of the data and the attitudes with which the data is treated. Although also grammar-based bots can gain more popularity as their successful generations are retweeted and shared on the readers' own timelines, the data presented by list-based bots is generally selected to encourage the readers to project their own meaning to the dry data while sharing their own anecdotes to their followers. When paired with even a simple boilerplate text template, the author of a list-based bot is able to change the reader's attitude towards the data, highlighting the absurdity that lies in, for example, scientifically categorised food names or words in a dictionary.

The reading of a bot occurs over time as more and more samples of the bot's output appear on a user's timeline. With more and more iterations of the bot's text-to-be-seen encountered by the reader, the more familiar the reader becomes of the details of the format as well as of the malleability of language in general. The characteristics and metrics looked at in this thesis are instinctive to the reader of the bot as they face the bots' updates directly on their timeline. If the bot remains updated and varied, the user is more likely to continue following the account. However, if the bot repeats the same kind of content from one update to another without any relevance or discovery to the reader, the updates of the bot will never be shared through retweets or quoted tweets and the following of the bot will quickly diminish. The framework used in the case studies of the previous chapter helps to formalise the metrics with which the interestingness and sustainability of a bot can be evaluated in a methodological manner.

* * *

The content generated by bots is present in the day-to-day life of the readers who follow bot accounts. The output of bots jumps out from the feed of the mundane, intention-typical language of other Twitter users. Bots can give their followers a respite from the human-generated content, the brand promotion and links shared on the user's timeline. In a way, bots can also be seen as an attempt to subvert Twitter's business model, which focuses on commodifying tweets by their value in creating an advertising profile of the users. As noted by Allison Parrish, content generated by bots is the Twitter-equivalent of skateboarding, where an existing platform is given a new use

when it is approached with unlimited, playful attitude.[1] Bots are an example of ludic dysfunction, as described by Marie-Laure Ryan, and in addition to the artistic values they embody, they can make "users aware of the codes and processes (technological, linguistic, cultural and cognitive) that regulate our social and mental life."[2] This awareness can result from a lot of sources: some bots are clearly critical of the formulaic language used in everyday life and many expose the limits of human imagination with their mechanical combinations of concepts. However, all bots are subversive in one particular way. With their relentless generation and posting of content, they also point out the widespread use of computer-generated content in the online environment: if an individual programmer can create a bot that generates content throughout the day at minimum cost, to what extent are all the other, more commercialised fields of life dominated by computer-generated content?

Bots are highly accessible to both readers and creators. As bots post their generations directly to the social network, they can be viewed in the same environment where people already spend a lot of their time. Discovery and reading of bots need not be an active process as a reader can encounter new bots through retweets by other users and, thus, the publication of bots is easy and does not rely on being listed on websites or online directories like other works of e-lit. There are also many open-source tools and tutorials available to interested creators free of charge, enabling a creator with the skill set of an average web user to create a bot in a few hours and publish it with a couple of clicks. These tools show that understanding the functioning of bots does not require the reader or the creator of bots to understand programming or the technologies running bots. Rather, the most important aspect in creating and reading bots is an understanding of the textual mechanics and the processes which turn a small set of initial data, boilerplate sentences and selected word lists into thousands and thousands of unique micronarratives, love letters, novel descriptions of emotions or ironic thinkpiece headlines.

* * *

Twitter bots, as well as other textual generators, often raise questions of authorship. Many are quick to claim that the poetry and stories generated by algorithms are evidence for machine creativity, as noted by Bootz in the introduction of his functional view, "Should the computer be considered an artificial ghost-writer? Such an idea might more quickly come to minds when dealing with generators, for in this case the reader, having been invited to take a part in it, clearly identifies that the writing is not completed by the author."[3] This belief, however, is dismissed by the theorists and makers mentioned above. It is possible to see from the application of Bootz's functional view into the process of creating bots that human creativity lies at the heart of the creative process while the generation of the texts-to-be-seen is merely a method to compile texts according to the author's idea of the work. Botmaking is not unlike writing poetry, although in botmaking the making of language turns into second-order making, the creation of a machine that creates stories, poems and evocative language. The computer is not the author as it merely executes the author's design without any innovation of its own. The framework described in this thesis takes into consideration the author's essential role in the creation of the author-text by looking behind the texts-to-be-seen in an attempt to understand the artistic choices made in all

---

1 Parrish, "Understanding Bots."

2 Ryan, "Between Play and Politics."

3 Bootz, "The Functional Point of View," 315.

phases of the creative process. Although the bot's output is random and sometimes even incomprehensible, the choices that lead to this variability never are.

Botmaking is an accessible craft with immediate results and easy platforms for publication. As noted, rather than requiring ready knowledge of computer programming, the creation of an intriguing and sustainable Twitter bot is more related to understanding generative textual mechanics, which have been understood and utilised already in works that precede Twitter, the internet and even digital computing. Examples like Strachey's love letter generator and Gysin's permutation poetry show how a complex surface does not necessarily mean a complex source. Rather, with skilful attention to the language and the rules that make up the author-text, careful curation of data sources and a basic understanding of textual mechanics and the history of textual generation, anyone can make a bot that, with a thoughtful, simple design, generates intriguing, artistic and novel ideas that one would not expect from a human author.

Bibliography

Works Cited

"About Wordnik." Wordnik. Accessed January 28, 2017. http://www.wordnik.com/about.

Aarseth, Espen J. *Cybertext: Perspectives on Ergodic Literature*. Baltimore: Johns Hopkins University Press, 1997.

———. "Nonlinearity and Literary Theory." In *Hyper / Text / Theory*. Edited by George P. Landow, 51–86. Baltimore: Johns Hopkins University Press, 1994.

Berens, Kathi Inman. "Tournedo Gorge" (Author's Statement). *Electronic Literature Collection* 3 (February, 2016). Accessed February 7, 2017. http://collection.eliterature.org/3/work.html?work=tournedo-gorge.

Bootz, Philippe. "The Functional Point of View: New Artistic Forms for Programmed Literary Works." *Leonardo* 32, no. 4 (1999): 307–16. Accessed May 18, 2016. doi: 10.1162/002409499553307.

Burroughs, William S. "Note on Vaudeville Voices." In *The Moderns: An Anthology of New Writing in America*, edited by Leroi Jones. New York: Corinth Books, 1963. Accessed October 15, 2016. http://www.writing.upenn.edu/~afilreis/88v/burroughs-cutup.html.

———. "The Cut Up Method." In *The Moderns: An Anthology of New Writing in America*, edited by Leroi Jones. New York: Corinth Books, 1963. Accessed October 15, 2016. http://www.writing.upenn.edu/~afilreis/88v/burroughs-cutup.html.

Campbell, Jim. "I Have Never Read The Bible." Jim Campbell. Accessed February 6, 2017. http://www.jimcampbell.tv/portfolio/objects/memory_works/i_have_never_read_the_bible/.

Compton, Kate. "I, Twitter Bot." Interview by Ira Flatow. *Science Friday*. Podcast audio. August 19, 2016. Accessed August 25, 2016. http://www.sciencefriday.com/segments/i-twitter-bot/.

———. "So You Want to Build a Generator." *Kate Compton* (Tumblr blog). February 22, 2016. Accessed October 2, 2016. http://galaxykate0.tumblr.com/post/139774965871/so-you-want-to-build-a-generator.

Cramer, Florian. *Words Made Flesh: Code, Culture, Imagination*. Rotterdam: Media Design Research, Piet Zwart Institute, Willem de Kooning Academy, Hogeschool Rotterdam, 2005. Accessed May 19, 2016. http://www.netzliteratur.net/cramer/wordsmadefleshpdf.pdf.

"Fred and George" (Editorial Statement). *Electronic Literature Collection* 3 (February, 2016). Accessed February 7, 2017. http://collection.eliterature.org/3/work.html?work=fred-and-george.

Gaboury, Jacob. "A Queer History of Computing: Part Three." *Rhizome* (blog), April 9, 2013. Accessed October 30, 2016. http://rhizome.org/editorial/2013/apr/09/queer-history-computing-part-three/.

"George Buckenham / Cheap Bots, Done Quick!" Abandon Normal Devices. Accessed February 1, 2017. http://www.andfestival.org.uk/events/george-buckenham/.

Glazier, Loss Pequeño. *Digital Poetics: The Making of E-Poetries*. Tuscaloosa: The University of Alabama Press, 2002.

Hayles, N. Katherine. *Electronic Literature: New Horizons for the Literary*. Notre Dame, IN: University of Notre Dame, 2008. PDF e-book.

———. *My Mother Was a Computer: Digital Subjects and Literary Texts*. Chicago: University of Chicago Press, 2005. PDF e-book.

Kazemi, Darius. "Corpora." Github. September 23, 2016. Accessed January 28, 2017. https://github.com/dariusk/corpora.

———. "I, Twitter Bot." Interview by Ira Flatow. *Science Friday*. Podcast audio. August 19, 2016. Accessed August 25, 2016. http://www.sciencefriday.com/segments/i-twitter-bot/.

———. "Tiny Subversions." Tiny Subversions. Accessed October 11, 2016. http://tinysubversions.com/.

Klink, Flourish. "Fred and George" (Author's Statement). *Electronic Literature Collection* 3 (February, 2016). Accessed February 7, 2017. http://collection.eliterature.org/3/work.html?work=fred-and-george.

Lutz, Theo. "Stochastic Texts." Translated by Helen MacCormack. Stuttgarter Schule. 2005. Accessed April 1, 2017. http://www.stuttgarter-schule.de/lutz_schule_en.htm. Originally published as "Stochastische Texte" in *Augenblick* 4, no. 1 (1959): 3–9.

Marino, Mark C. "Critical Code Studies." *Electronic Book Review*, December 4, 2006. Accessed February 29, 2016. http://electronicbookreview.com/thread/electropoetics/codology.

Molloy, Mark. "*Record Scratch* *Freeze Frame*: The Hilarious Meme Taking Over the internet." *The Telegraph*, September 2, 2016. Accessed January 12, 2017. http://www.telegraph.co.uk/news/2016/09/02/record-scratch-freeze-frame-the-hilarious-meme-taking-over-the-i/.

Montfort, Nick. "Conceptual Computing and Digital Writing." Forthcoming in *Postscript: Writing After Conceptual Art*, edited by Andrea Andersson. Toronto: University of Toronto Press, 2017. Accessed April 1, 2017. http://hdl.handle.net/1721.1/92876.

———. "Taroko Gorge" (Author's Statement). *Electronic Literature Collection* 3 (February, 2016). Accessed February 6, 2017. http://collection.eliterature.org/3/work.html?work=taroko-gorge.

Orlean, Susan. "Man and Machine." *The New Yorker*, February 10, 2014. Accessed January 10, 2017. http://www.newyorker.com/magazine/2014/02/10/man-and-machine-2.

Parrish, Allison. "N-Grams and Markov Chains." Decontextualize. Accessed June 30, 2016. http://www.decontextualize.com/teaching/rwet/n-grams-and-markov-chains/.

———. "Understanding Bots." Presentation at ELO 2016, University of Victoria, June 22, 2016. PDF presentation slides and notes. Accessed June 19, 2016. https://s3.amazonaws.com/aparrish/procedure-vs-procedure-elo2016.pdf.

Pipkin, Katie Rose. "About a Bot: Interview with Katie Rose Pipkin." Interview by Taina Bucher. *Furtherfield*, December 7, 2015. Accessed March 30, 2017. http://furtherfield.org/features/interviews/about-bot-interview-katie-rose-pipkin.

Reed, Nora. "How and Why @nerdgarbagebot Works." *Gusty Winds May Exist* (blog), January 10, 2016. Accessed October 2, 2016. http://barrl.net/2801.

———. "The Official @Thinkpiecebot FAQ." *Gusty Winds May Exist* (blog), September 6, 2015. Accessed October 2, 2016. http://barrl.net/2748.

Rettberg, Scott. "Tokyo Garage" (Author's Statement). *Electronic Literature Collection* 3 (February, 2016). Accessed February 7, 2017. http://collection.eliterature.org/3/work.html?work=tokyo-garage.

Rodley, Chris. “Introducing Chris Rodley.” Interview by Sarah Blaszczok. *Abandon Normal Devices* (blog), April 12, 2016. Accessed October 2, 2016. http://www.andfestival.org.uk/blog/introducing-chris-rodley-the-art-of-bots/.

Rodley, Chris and Andrew Burrell. “On the Art of Writing with Data.” In *The Future of Writing*, edited by John Potts, 77–89. Basignstoke: Palgrave Macmillan, 2014. doi: 10.1057/9781137440402.0012.

Rodley, Chris and Ali Rodley. “In Conversation with Chris and Ali Rodley: The Creators of the Magical Realism Bot.” Interview by Matthew Spencer. *Asymptote*, March 30, 2016. Accessed October 2, 2016. http://www.asymptotejournal.com/blog/2016/03/30/in-conversation-with-chris-and-ali-rodley-the-creators-of-the-magical-realism-bot/.

Ryan, Marie-Laure. “Between Play and Politics: Dysfunctionality in Digital Art.” *Electronic Book Review*, March 3, 2010. Accessed October 23, 2015. http://www.electronicbookreview.com/thread/imagenarrative/diegetic.

Sample, Mark. “A Protest Bot Is a Bot So Specific You Can’t Mistake It for Bullshit: A Call for Bots of Conviction.” *Medium*, May 30, 2014. Accessed October 2, 2016. https://medium.com/@samplereality/a-protest-bot-is-a-bot-so-specific-you-cant-mistake-it-for-bullshit-90fe10b7fbaa.

“Scholars Contemplate the Irish Beer” (Editorial Statement). *Electronic Literature Collection* 3 (February, 2016). Accessed February 6, 2017. http://collection.eliterature.org/3/work.html?work=scholars-contemplate-the-irish-beer.

Soini, Timo. “Haastamalla horjutetaan – punavihreät kyykkäsivät.” *Ploki* (blog), May 17, 2016. Accessed March 28, 2017. http://timosoini.fi/2016/05/haastamalla-horjutetaan-punavihreat-kyykkasivat/.

———. “Lässytys lyö takaisin.” *Ploki* (blog), June 5, 2016. Accessed March 28, 2017. http://timosoini.fi/2016/06/lassytys-lyo-takaisin/.

———. “Olenko minä nero?” *Ploki* (blog), June 19, 2016. Accessed March 28, 2017. http://timosoini.fi/2016/06/olenko-mina-nero/.

———. “Politiikka on ihanaa.” *Ploki* (blog), June 12, 2016. Accessed March 28, 2017. http://timosoini.fi/2016/06/politiikka-on-ihanaa/.

Strickland, Stephanie. “Writing the Virtual: Eleven Dimensions of E-Poetry.” *Leonardo Electronic Almanac* 14, no. 5–6 (Aug. & Sept. 2006). Accessed May 19, 2016. http://www.leoalmanac.org/wp-content/uploads/2012/09/06Writing-the-Virtual-Eleven-Dimensions-of-E-Poetry-by-Stephanie-Strickland-Vol-14-No-5-6-September-2006-Leonardo-Electronic-Almanac.pdf.

“Taroko Gorge Remixes.” *Electronic Literature Collection* 3 (February, 2016). Accessed February 6, 2017. http://collection.eliterature.org/3/collection-taroko.html.

Taylor, Scott. “Alison Knowles, James Tenney and the *House of Dust* at CalArts.” *24700: News from California Institute of Arts* (blog), September 10, 2009. Accessed November 8, 2016. http://blog.calarts.edu/2009/09/10/alison-knowles-james-tenney-and-the-house-of-dust-at-calarts/.

“Ulkoministeri Soini kirjoitti Nytin olevan ‘kiviterapiakommunismia’, mitä se tarkoittaa?” *Nyt*, June 6, 2016. Accessed March 28, 2017. http://www.hs.fi/nyt/art-2000002904869.html.

Wardrip-Fruin, Noah. “Authoring Processes.” *Grand Text Auto* (blog), January 23, 2008. Accessed June 30, 2016. https://grandtextauto.soe.ucsc.edu/2008/01/23/ep-12-authoring-processes/.

———. “Five Elements of Digital Literature.” In *Reading Moving Letters: Digital Literature in Research and Teaching*, edited by Peter Gendolla, Roberto Simanowski and Jörgen Schäfer, 29–57. Bielefeld: Transcript, 2010.

"What is E-Lit?" Electronic Literature Organization. Accessed October 7, 2016. http://eliterature.org/what-is-e-lit/.

Winder, William. "Robotic Poetics." In *A Companion to Digital Humanities*, edited by Susan Schreibman, Ray Siemens and John Unsworth. Oxford: Blackwell, 2004. Accessed May 12, 2016. http://digitalhumanities.org/companion/view?docId=blackwell/9781405103213/9781405103213.xml&chunk.id=ss1-4-11.

Zhang, Jia. "Introducing censusAmericans, A Twitter Bot For America." *FiveThirtyEight*, July 24, 2015. Accessed January 12, 2017. https://fivethirtyeight.com/datalab/introducing-censusamericans-a-twitter-bot-for-america/.

Bots

Bhatnagar, Ranjit. *Pentametron* (*@pentametron*). Accessed January 12, 2017. https://twitter.com/pentametron.

Compton, Kate. *Tiny Space Adventure* (*@TinyAdv*). Accessed January 12, 2017. https://twitter.com/TinyAdv.

Twitter user *@dullthud*. *Vennsplain* (*@vennsplain*). Accessed January 15, 2017. https://twitter.com/vennsplain.

Twitter user *@duukkis*. *Edesmenneet* (*@EdesmenneetPrh*). Accessed January 15, 2017. https://twitter.com/EdesmenneetPrh.

———. *Jaxubotti* (*@jaxubotti*). Accessed January 31, 2017. https://twitter.com/jaxubotti.

———. *Se Wsi Testamentti* (*@testamenttwit*). Accessed January 15, 2017. https://twitter.com/testamenttwit.

Glosson, Amanda. *Tiny Neighbor* (*@tinyneighbor*). Accessed January 12, 2017. https://twitter.com/tinyneighbor.

———. *Tiny Spires* (*@tinyspires*). Accessed January 12, 2017. https://twitter.com/tinyspires.

Every Word Is Gay (*@everywordisgay*). Accessed January 15, 2017. https://twitter.com/everywordisgay.

Happnin' in Hastings (*@What_Hastings*). Accessed October 13, 2016. https://twitter.com/What_Hastings.

Higgins, Parker. *Trains Botting* (*@choochoobot*). Accessed October 13, 2016. https://twitter.com/choochoobot.

Twitter user *@hugovk*. *Every Finnish Number* (*@@EveryFinnishNo*). Accessed January 15, 2017. https://twitter.com/everyfinnishno.

———. *Every Finnish Word* (*@kaikkisanat*). Accessed January 15, 2017. https://twitter.com/kaikkisanat.

Kazemi, Darius. *Two Headlines* (*@TwoHeadlines*). Accessed January 12, 2016. https://twitter.com/TwoHeadlines.

Koola, Timo. *Finnegans Wake* (*@finnegansreader*). Accessed January 15, 2017. https://twitter.com/finnegansreader.

———. *Twiittejä Suomesta* (*@prklsuomi*). Accessed January 12, 2017. https://twitter.com/prklsuomi.

———. *Ulysses Reader* (*@UlyssesReader*). Accessed January 15, 2017. https://twitter.com/UlyssesReader.

Lampi, Ville Matias. *Kaikki on pahaa* (*@kaikkionpahaa*). Accessed January 15, 2017. https://twitter.com/kaikkionpahaa.

———. *#fillarikommunismi* (*@soinismibot*). Accessed January 28, 2017. https://twitter.com/soinismibot.

Twitter user *@MaybeAlexL*. *Gay Update* (*@GayUpdateBot*). Accessed January 28, 2017. https://twitter.com/GayUpdateBot.

Moore, Brian. *Freeze Frame Bot* (*freezeframebot*). Accessed January 12, 2017. https://twitter.com/freezeframebot.

Nasser, Ramsay. *Everyunicode* (*@everyunicode*). Accessed January 15, 2017. https://twitter.com/everyunicode.

Neon Clock Tweet Bot (*@NixieBot*). Accessed January 31, 2017. https://twitter.com/NixieBot.

Parrish, Allison. *Everyword* (*@everyword*). Accessed January 12, 2017. https://twitter.com/everyword.

———. *The Ephemerides* (*@the_ ephemerides*). Accessed January 12, 2017. https://twitter.com/the_ephemerides.

Pipkin, Katie Rose. *Feelings.js* (*@feelings_ js*). Accessed April 2, 2017. https://twitter.com/feelings_js.

———. *Tiny Star Fields* (*@tiny_ star_ field*). Accessed January 12, 2017. https://twitter.com/tiny_star_field.

Pusa, Petri. *Signe Brander Bot* (*@signebrander*). Accessed January 12, 2017. https://twitter.com/signebrander.

Päivän annos (*@PaivanAnnos*). Accessed April 2, 2017. https://twitter.com/PaivanAnnos.

Rare Lifestyles. *Fuck Every Word* (*@fuckeveryword*). Accessed January 15, 2017. https://twitter.com/fuckeveryword.

Reed, Nora. *Lady Products* (*@lady_ products*). Accessed April 10, 2017. https://twitter.com/lady_products.

———. *Man Products* (*@man_ products*). Accessed April 10, 2017. https://twitter.com/man_products.

———. *Thinkpiece Bot* (*@thinkpiecebot*). Accessed January 12, 2017. https://twitter.com/thinkpiecebot.

Rodley, Chris and Ali Rodley. *Magic Realism Bot* (*@MagicRealismBot*). Accessed January 12, 2017. https://twitter.com/MagicRealismBot.

Sample, Mark. *NSA PRISM* (*@NSAPRISMbot*). Accessed January 26, 2017. https://twitter.com/NSA_PRISMbot.

Strebel, Joshua. *Willy Shakes* (*@IAM_ SHAKESPEARE*). Accessed January 26, 2017. https://twitter.com/IAM_SHAKESPEARE.

StubbBot (*@StubbBot*). Accessed January 12, 2017. https://twitter.com/StubbBot.

Summers, Ed. *Congress-Edits* (*@congressedits*). Accessed October 13, 2016. https://twitter.com/congressedits.

Twitter user *@vogon*. *Every Color* (*@everycolorbot*). Accessed January 15, 2017. https://twitter.com/everycolorbot.

Zhang, Jia. *CensusAmericans* (*@censusAmericans).* Accessed January 12, 2017. https://twitter.com/censusAmericans.

Artworks

Andrews, Jim. *Stir Fry Texts*, 1999–. In *Electronic Literature Collection* 1 (October 2006). Accessed March 21, 2017. http://collection.eliterature.org/1/works/andrews__stir_fry_texts.html.

Berens, Kathi Inman. *Tournedo Gorge*, 2012. In *Electronic Literature Collection* 3 (February, 2016). Accessed February 6, 2017. http://collection.eliterature.org/3/work.html?work=tournedo-gorge.

Campbell, Jim. *I Have Never Read the Bible*, 1995. Installation: Webster's dictionary, speaker, custom electronics, 71 x 15 x 6 inches.

Chamberlain, William. *The Policeman's Beard Is Half-Constructed: Computer Poetry and Prose by Racter*. New York: Warner Books, 1984. August 10, 2014. Accessed November 2, 2016. https://archive.org/details/pdfy-T3abGAQ80ecd63PL.

Geniwate, and Brian Kim Stefans. *When You Reach Kyoto*. In *Electronic Literature Collection* 1 (October 2006). Accessed March 21, 2017. http://collection.eliterature.org/1/works/geniwate__generative_poetry.html.

Glazier, Loss Pequeño. *Mouseover*, 1998. Accessed April 23, 2017. http://wings.buffalo.edu/epc/authors/glazier/viz/mouseover/mouseover.html.

Gomringer, Eugen. "3 variationen zu kein fehler im system." In *Konkrete Poesie*. Edited by Eugen Gomringer. Stuttgart: Reklam, 1972.

Gysin, Brion. "I Am That I Am," 1959. Sound recording. Accessed October 15, 2016. https://ubusound.memoryoftheworld.org/gysin_brion/Mektoub/Gysin-Brion_Mektoub_05-I-Am-That-I-Am.mp3.

———. "In the Beginning Was the Word," 1969. Sound recording. Accessed October 15, 2016. https://ubusound.memoryoftheworld.org/gysin_brion/Mektoub/Gysin-Brion_Mektoub_21-In-The-Beginning-Was-The-Word.mp3.

———. "No Poets Don't Own Words." Sound recording. Accessed October 15, 2016. https://ubusound.memoryoftheworld.org/gysin_brion/Mektoub/Gysin-Brion_Mektoub_10-No-Poets-Don't-Own-Words.mp3.

Klink, Flourish. *Fred and George*, 2012. In *Electronic Literature Collection* 3 (February, 2016). Accessed February 6, 2017. http://collection.eliterature.org/3/work.html?work=fred-and-george.

Knowles, Allison and James Tenney. *A House of Dust*, 1967. Accessed November 8, 2016. http://beinecke.library.yale.edu/about/blogs/room-26-cabinet-curiosities/2010/02/15/house-dust.

LaPorte, Stephen, and Mahmoud Hashemi. *Listen to Wikipedia*, 2013. Accessed March 21, 2017. http://listen.hatnote.com.

Lutz, Theo. "Stochastische Texte." *Augenblick* 4, no. 1 (1959): 3–9.

Montfort, Nick. *Flow My Tears*, 2015. Accessed February 7, 2017. http://nickm.com/poems/flow_my_tears.html.

———. *Taroko Gorge*, 2009. In *Electronic Literature Collection* 3 (February, 2016). Accessed February 6, 2017. http://collection.eliterature.org/3/work.html?work=taroko-gorge.

Montfort, Nick and Stephanie Strickland. *Sea and Spar Between*, 2010. Accessed February 7, 2017. http://www.saic.edu/webspaces/portal/degrees_resources/departments/writing/DNSP11_SeaandSparBetween/index.html.

Queneau, Raymond. *Cent mille milliards de poèmes*. Gallimard: Paris, 1961.

———. “100,000,000,000,000 Poems.” Translated by Stanley Chapman. In *Oulipo Compendium*, edited by Harry Mathews and Alastair Brotchie, 15-33. London: Atlas Press, 1998. Reproduced in *The New Media Reader*, edited by Nick Montfort and Noah Wardrip-Fruin, 149–169. Cambridge, MA: The MIT Press, 2003. Accessed March 16, 2017. http://nickm.com/classes/the_word_made_digital/2014_spring/12-oulipo-p.pdf.

Rettberg, Scott. *Tokyo Garage*, 2009. In *Electronic Literature Collection* 3 (February, 2016). Accessed February 7, 2017. http://collection.eliterature.org/3/work.html?work=tokyo-garage.

Sample, Mark. *House of Leaves of Grass*, 2013. Accessed February 7, 2017. http://fugitivetexts.net/houseleavesgrass/index.html.

Strachey, Christopher. *Love Letters*, 1952.

Tzara, Tristan. “How to Make a Dadaist Poem,” 1920. Accessed October 13, 2016. http://www.writing.upenn.edu/~afilreis/88v/tzara.html.

Young-Hae Chang Heavy Industries. *Dakota*, 2001. Accessed April 23, 2017. http://www.yhchang.com/DAKOTA.html.