AI-Generated Literature and the Vectorized Word

by

Judy Heflin

B.A. Comparative Literature and Culture, Yonsei University (2015)
Submitted to Program in Comparative Media Studies/Writing in
Partial Fulfillment of the Requirements for the Degree of…

Master of Science in Comparative Media Studies
at the Massachusetts Institute of Technology
May 2020

Signature of Author:

Department of Comparative Media Studies
May 8, 2020

Certified By:

Nick Montfort
Professor of Digital Media
Thesis Supervisor

Certified By:

Lillian-Yvonne Bertram
Assistant Professor of Creative Writing
Director, MFA in Creative Writing, UMass Boston
Thesis Supervisor

Accepted By:

Eric Klopfer
Professor of Comparative Media Studies
Director of Graduate Studies

AI-Generated Literature and the Vectorized Word
by
Judy Heflin

Submitted to the Program in Comparative Media Studies/Writing
on May 8, 2020 in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Comparative Media Studies.

ABSTRACT

This thesis focuses on contemporary AI-generated literature that has been traditionally published in the form of a printed book, labeled and interpreted as something written by "artificial intelligence," and that necessarily depends on vector representations of linguistic data. This thesis argues that AI-generated literature is not a monolithic practice that erases the role of the author, but rather encompasses a diversity of practice that includes divergent artistic and writerly perspectives. Through an in-depth look at three books of contemporary AI-generated literature and discussions with their human authors, this thesis details the authorial and writerly labor throughout the stages of datafication, vectorization, generation, and pagination in order to categorize the writerly practices that are involved in the creation of this type of work. This thesis also considers how these practices are preceded by "analog" types of writing, compares divergent authorial perspectives, and discusses ways of reading AI-generated literature, including a material analysis of how AI-generated text interacts with the printed book, how authors and publishers use paratextual elements to guide readings, along with additional points of entry for analyzing literary AI-generated texts.

Thesis Supervisor: Nick Montfort
Title: Professor of Digital Media

Thesis Supervisor: Lillian-Yvonne Bertram
Title: Assistant Professor of Creative Writing
Director, MFA in Creative Writing, UMass Boston

ACKNOWLEDGEMENTS

Institutional: This thesis could not have been written without the guidance of Nick Montfort and Lillian-Yvonne Bertram. I am so grateful for their mentorship and constantly inspired by their creative and scholarly work. Thank you to Nick for supervising this thesis and taking me on as both a research and teaching assistant in what people close to me have begun referring to as the “wax on, wax off” of computational literature. Over the past two years I’ve learned more than I ever thought I could, and more than I ever knew there was to know. Highlights from the Trope Tank include: learning to write a BASIC program on a ZX Spectrum and save it to tape, learning how to properly use a tape recorder in the first place, some of the nerdiest rap battles I have ever had the pleasure of witnessing, and a glorious collection of computer-generated books in print. My infinite gratitude goes to Allison Parrish, Ross Goodwin, and David Jhave Johnston for entertaining all of my questions, allowing me to publish our conversations, and inspiring me in so many ways.

Personal: Thank you to my mom, for giving me such a welcoming place to finish my thesis during the global pandemic and for the late-night espressos and unconditional love. Thank you to Dave whose rigorous work ethic and commitment to the grueling process of digging a basement by hand motivated me to continue writing. Thanks to my dad for supporting my education in so many ways, for the hours of Myst-playing and puzzle-solving, and for teaching me that I could survive at a place like MIT. Thanks to Julia, whose visits, memes, Duo calls, and friendship I could not have survived the last two years without. A big thank you to Marro, for getting me through my deepest pits of despair and self-doubt during graduate school and believing in me to the point of delusion. Thank you to my amazing cohort: Anna, Annie, Ben, Han, Iago, Lizzie, and Sam. I’m so proud to know every one of you.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY OF TERMS

Artificial Intelligence (AI): Computer programs with the ability to learn and reason like humans

Convolutional Neural Network (CNN): A type of neural network that is ideal for processing image data and handling classification tasks

Computer-generated literature: Literary text that has been generated in some way through running a computer program

Corpus: A collection of written texts used as source material or data for a work of computer-generated literature

Corpora: Plural of corpus

Dense Vector: A vector in which most of the indices are nonzero numbers

Dimensionality: The number of dimensions in a vector space. We can easily visualize 2-dimensional or 3-dimensional spaces, but the text-generation systems that we will look at utilize very high dimensions that are not easy to visualize.

Embedding: The process by which text is given numerical representation in a vector space

Language Model (LM): A computational representation of language that encodes some sort of linguistic understanding

Long-short-term-memory (LSTM): A type of recurrent neural network with the added ability to store past information at each processing step in order to improve modeling and prediction of sequences of data

Machine Learning (ML): Algorithms that are able to draw conclusions from data without being explicitly programmed

Neural Network: A computational model inspired by the structure of the human brain that can automatically detect patterns in large amounts of data

Natural Language Generation (NLG): A subfield of natural language processing that focuses on computationally generating text

Natural Language Processing (NLP): A subfield of artificial intelligence that focuses on the computational manipulation of linguistic data

Recurrent Neural Network (RNN): A type of neural network that is used to model, classify, and predict sequences of data such as text

Rule-based System: A text-generation system wherein linguistic understanding and relationships are explicitly encoded

Scalar: A quantity with magnitude but no direction that can be measured by a real number

Seed: Data that initializes some sequence of generation powered by a neural network

Sparse Vector: A vector in which most of the indices are zero

Token: Every word in a corpus, including repeated words

Training: The computational process in which the weights and biases of a neural network are changed in order to automatically encode meaningful patterns in data

Type: Every individual word in a corpus, not including repeated words

Vector: A quantity with magnitude and direction, a sequence of numbers used to describe a point in a high-dimensional space

Vector Space: A high-dimensional space that includes a set of vectors that represent data in some dataset. In this case, because we are looking at textual data, the vector spaces in the systems that generate the primary sources we will look at automatically encode certain types of linguistic relationships, sometimes phonetic, and sometimes semantic.

1. AI-GENERATED LITERATURE

INTRODUCTION

In the summer of 2018, the press release for *1 the Road,* a computational creative writing project by Ross Goodwin, claimed that the work was the "first real book written by an AI." The blurb continues: "*1 the Road* imposes a new reflection on the place and authority of the author in a new era of machines."[1] But what is meant here by AI? Why is the most interesting question posed by this book one of authorship? What marks the beginning of this *new era of machines*? 25 years before *1 the Road*, in 1993, another AI-generated book was published claiming to be the first of its kind. The jacket copy of *Just This Once: A Novel Written by a Computer Programmed to Think Like the World's Bestselling Author* reads; "Once in a lifetime, an author writes a book that sets the literary world on its ear. But never before has that author been a computer."[2] Ten years prior to this, in 1984, Grand Central Publishing issued *The Policeman's Beard is Half Constructed.* The authorial credit is given to its generating program, RACTER, and the bright red title page boasts that it is "the first book ever written by a computer."[3] This is also not true. The earliest book of computer-generated literature that has been documented in the Trope Tank's bibliography of computer-generated books in print was published in Quebec in 1964.[4] It was an experiment in computational linguistics more than a work of poetry by Jean Baudot called *A Writing Machine*. Artists and authors have been working with computational creative writing systems since as early as the 1950s. Decade after decade the outputs of these works are sold as

[1] Jean Boîte Éditions, *A gonzo AI takes the road after Kerouac to give us the first novel written by a machine* (September 2018), Press release. http://www.desrimais.com/FTP/JBE/00-CP_1_the_Road_EN.pdf.

[2] Scott French, *Just This Once* (New York, NY: Carol Publishing Group, 1993), Jacket copy.

[3] William Chamberlain and Thomas Etter *The Policeman's Beard is Half-constructed: Computer Prose and Poetry*, (New York, NY: Warner Software/Warner Books, 1984), Cover copy.

[4] Nick Montfort, "Computer-Generated Books," Post Position, October 31, 2019. https://nickm.com/post/2016/07/computer-generated-books/comment-page-1/.

novelty items, with press material that suggests that writing as a practice is on the verge of extinction, and that the author as we know it is dead.

When I spoke to Ross Goodwin, the author of *1 the Road*, or as the book specifies, the writer of the writer of *1 the Road*, he told me that he argued with his publisher about claiming that his book was the first of its kind at all. Although the publishers insisted that it was a defensible claim and that it would help the book sell, he did manage to convince them to put the statement about the book being any kind of first on the temporary sleeve, rather than the permanent cover.[5] This small negotiation is a great display of the complex array of human labor and perception that dictates the life of computer-generated books: from experimental artists trying to push the boundaries of literature, to the monetary realities of the publishing industry; from the software developers creating open source frameworks that make this kind of independent work possible, to investments and funding from larger corporations; from the curation and maintenance of large data sets of literary work, to the long history of authors who have written the words of a collective, mechanized voice.

Much is lost when the only questions raised by computer-generated books come from the push for novelty and sensationalism, especially in an age so intrinsically entwined with the logic of computation. There are many interesting and fruitful questions to ask other than what it means for a machine to be an author, which is a conversation that often overshadows other areas of inquiry regarding these texts. For example, in what ways has our culture embraced the logic of computation? How can we use the patterns and techniques of algorithms to say something about the world as it is? How are human authors currently doing this with computation? Why are we so

[5] Ross Goodwin (Writer of writer of *1 the Road*), in discussion with the author, February 25, 2019. See Appendix B.

eager to anthropomorphize a machine? What can literary explorations of data curation, structures, and analysis reveal about the contemporary quantified self? How can these computational ontological tools be absorbed into the realm of literature? Why aren't more authors using computation and computationally-mediated understandings of the self to generate literature? How are the characters in a story reckoning with a digital world? How are they in conversation with their own data? Would your story or poetic project be amplified by using specific computational techniques in the same way you would strategically implement a literary device to achieve a certain tone or rhythm or shape? This thesis does not try to answer all of these questions, but does try to move toward a better understanding of computer-generated literature by taking them seriously, rather than focusing on novelty or sensationalism.

Computer-generated writing has a history of making "most men in letters arouse tearful laments punctuated by cries of execration," as Italo Calvino describes in his 1967 lecture titled *Cybernetics and Ghosts*.[6] In Calvino's speculations on cybernetics and literature, he imagines a literary automaton that can perfectly follow all of the rules of classical literature as set out by structuralists and linguists, saying that "the use so far made of machines by the literary avant-garde is still too human."[7] The proliferation of computational writing experiments was well underway in the 1960s with the work of artists, programmers, and writers, such as John Cage, Jean Baudot, and Allison Knowles. Fifty years later in the development of computation, artificial intelligence, and computer-generated literature, there are now similar critiques and fear-based speculations about writing automatons.[8] However, Calvino would probably also

[6] Italo Calvino, "Cybernetics and Ghosts" in *The Uses of Literature: Essays* (Harcourt Brace & Company, 1987), 14.
[7] Calvino, *Cybernetics and Ghosts*, 13.
[8] Steven Poole, "The Rise of Robot Authors: Is the Writing on the Wall for Human Novelists?" *The Guardian*, March 25, 2019.

categorize the practice and poetics of computer-generated and AI-generated writing today as "still too human."

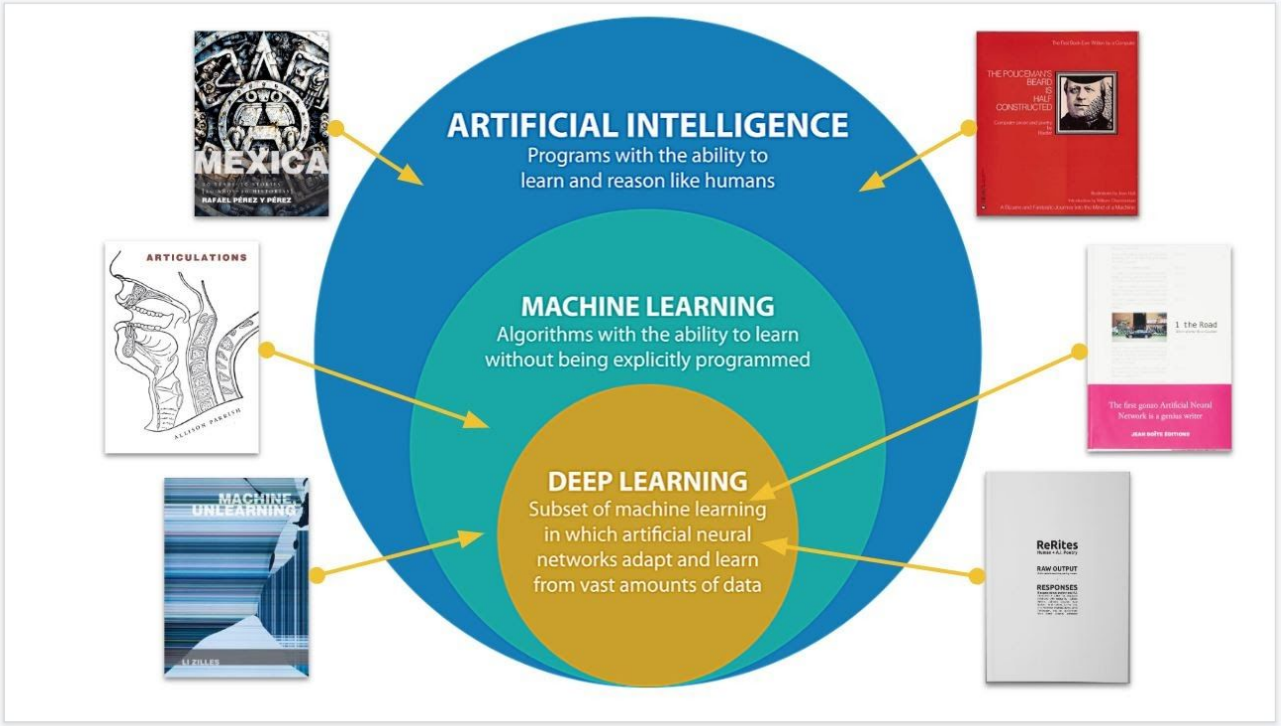

ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data
MEXICA
RAFAEL PÉREZ Y PÉREZ
ARTICULATIONS
ALLISON PARRISH
MACHINE UNLEARNING
LI ZILLES
THE POLICEMAN'S BEARD IS HALF CONSTRUCTED
1 the Road
The first gonzo Artificial Neural Network is a genius writer
JEAN BOÎTE ÉDITIONS
ReRites
RAW OUTPUT
RESPONSES

Figure 1. *Works of printed AI-generated literature, categorized by their text-generation techniques, and laid out on a diagram defining AI, ML, and deep learning.*[9]

The computer-generated book collection in the Trope Tank at MIT is home to almost 100 printed works of computational literature. Among those, many are labeled as AI-generated or use algorithms that are crucial in the history of AI. If you privilege the definition of AI as the discipline of computationally modeling human intelligence, then the book *MEXICA* would be a great example.[10] If you prioritize key algorithms in the history of AI and their creative applications, then *A Noise Such as a Man Might Make* would be a generative addition to the

https://www.theguardian.com/books/2019/mar/25/the-rise-of-robot-authors-is-the-writing-on-the-wall-for-human-novelists.

[9] "The Difference between Artificial Intelligence, Machine Learning and Deep Learning," *Data Catchup*, May 27, 2019. https://datacatchup.com/artificial-intelligence-machine-learning-and-deep-learning/.

[10] Rafael Pérez y Pérez, *Mexica: 20 Years-20 Stories [20 años-20 Historias]* (Denver: Counterpath, 2017).

discussion.[11] Although these aspects of AI are important, I found that I was interested in exploring contemporary representations of AI, widely perceived today as machine learning and deep learning models that rely heavily on statistics and linear algebra. I wanted to know what was unique about these models, especially in reference to their relationship with text, and what sort of writerly decisions and practices emerged from these characteristics. As a result of this interest, I soon became focused on the representation of language that makes the textual applications of these technologies possible: embeddings, or vector representations of textual data. The vectorization of language, or the process of placing linguistic data into a point in a high-dimensional space, is a way to represent that data numerically, which is a necessary step for making language legible to machine learning systems.[12]

This thesis will not be a holistic overview of authorial practices related to AI-generated writing nor a comprehensive history that compares AI-generated work to other computational modes of text generation, but a poetic investigation into the specific affordances of the vectorized word as it relates to writing, reading, the material book, and cultural perceptions of AI today. For the purposes of this thesis, I am limiting my scope of analysis to works of writing that have been traditionally published in the form of a printed book, labeled and interpreted as something written by "artificial intelligence," and that necessarily depend on vector representations of linguistic data. These three qualifications correspond with features that I see as crucial to the "writing" process of AI-generated literature: the material poetics of a literary AI system, how the work is conceptualized by and for the literary world, and how the work is developed given cultural conversations around AI at the time. There is enough of this limited sort

[11] Milton Läufer, *A Noise Such as a Man Might Make* (Denver: Counterpath, 2018).
[12] Allison Parrish, "Experimental Creative Writing with the Vectorized Word," YouTube, Strange Loop, September 30, 2017. https://www.youtube.com/watch?v=L3D0JEA1Jdc.

of literature to study in an analytical and comparative way, and as the barrier to entry for working with AI text-generation systems continues to lower, more will surely be published.

AI-generated literature has a long history of interesting primary sources to analyze and is actively being explored and expanded today. While many works of AI-generated literature have been created throughout the years, the books that are labelled as being written by AI today are mainly generated by text-generation algorithms that can automatically encode meaningful linguistic relationships in vector representations of data. Additionally, I argue that AI-generated literature is not a monolithic practice that erases the role of the author, but that a diversity of practice exists, and at times includes divergent artistic and writerly perspectives. Chapter 2 provides an in-depth look at three books of contemporary AI-generated literature, detailing the authorial and writerly labor throughout the stages of datafication, vectorization, generation, and pagination. Chapter 3 categorizes the writerly practices that are involved in the creation of AI-generated literature, considering how these practices are preceded by “analog” types of writing, and compares divergent authorial perspectives. Chapter 4 discusses ways of reading AI-generated literature, including a material analysis of how AI-generated text interacts with the printed book, how authors and publishers use paratextual elements to guide readings, along with additional points of entry for analyzing literary AI-generated texts.

CONCEPTS AND TERMS

This is not meant to be a thorough and in-depth technical explanation of state-of-the-art natural language generation techniques, but an overview of simplified, key vocabulary words that are necessary for a high-level understanding of the computational processes at work as discussed in this thesis, which focuses on creative text-generation systems that utilize natural language generation (NLG) techniques to computationally generate sequences of text. This category of literature can more simply be called computer-generated literature, which is some sort of literary output that is generated by running a computer program. NLG is a subfield of natural language processing (NLP), which is a subfield of artificial intelligence (AI). AI is a very broad category of computer science that focuses on programs with the ability to learn and reason like humans, while NLP is the branch of AI that is concerned with computational representations and understandings of language. NLG is a more narrow category of NLP that emphasizes novel and improved methods for computationally generating text. Historically, the two main approaches to NLG are rule-based (explicitly encoded) and machine learning-based (not explicitly encoded) computational models of language.[13] A language model, for the purposes of this thesis, is defined as any computational representation of language that encodes some sort of linguistic understanding. Although there are many works of AI-generated literature that utilize rule-based systems to generate text, this thesis focuses on the creative and writerly uses of machine learning, or computational systems that can automatically determine linguistic relationships and meaningful representations of language without explicitly programming rules of, for example, grammar or semantics. The computational representation of linguistic data that makes text readable to a machine learning program is the vector representation.

In the more scientific sense, a scalar is a quantity with only magnitude and no direction that can be measured by a real number, such as temperature or time. A vector is a quantity that has magnitude but also direction, and can be represented as a sequence of numbers used to describe a point in space. Some real world examples of vector quantities include force or

[13] Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei, "Context-aware Natural Language Generation with Recurrent Neural Networks," Cornell University, November 29, 2016. https://arxiv.org/abs/1611.09900

momentum. While vector representations of linguistic data don't necessarily have qualities that demonstrate magnitude or direction in physical space, their placement in an imaginary vector space introduces the notion of similarity among data points.[14]

All text on a computer has some sort of machine-readable representation. The American Standard Code for Information Interchange (ASCII) is one example of how text is represented numerically. Each letter in the ASCII character encoding standard is represented as a unique pattern made up of eight bits, or eight ones and zeros. For example, the acronym for Comparative Media Studies, CMS, would be represented as this ASCII encoding: "01000011 01001101 01010011." The simplest way to encode textual data in the form of vectors at the word level is through a process called one-hot encoding, in which each word is represented as a vector that is the length of the vocabulary size with all indices set to zero, other than the unique index-marker of the word, which is "switched on" to 1.[15]

Comparative	Media	Studies
[1, 0, 0]	[0, 1, 0]	[0, 0, 1]

This is an example of a sparse vector representation, in which the majority of indices in the vector are zeroes. While this type of vector is useful for categorizing unique types of data, it does not encode any notion of linguistic understanding or similarity. All that this representation can tell you about about these words is their index in a given dataset. It is also necessary at this point to make the quick distinction between tokens and types. If you are counting tokens in a dataset, then duplicate copies of the same data point are counted individually, whereas when counting types, each unique data point is counted, and repetitions are not considered. For

[14] Sebastian Ruder, "On Word Embeddings - Part 1," October 6, 2019. https://ruder.io/word-embeddings-1/.

[15] Jason Brownlee, "Why One-Hot Encode Data in Machine Learning?" Machine Learning Mastery, April 27, 2020. https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

example, ignoring punctuation, consider the following sentences: “I sit. I stand.” At the word-level, we have four tokens (“I,” “sit,” “I,” and “stand”) and three types (“I,” “sit,” and “stand”). If we were to encode this three-type “corpus” into a one-hot vector space, each type would be the same similarity and there would be no meaningful difference between sitting or standing. A corpus is a collection of written texts used as source material or data for a work of computer-generated literature.

This process of giving a unit of linguistic data (a word, for example) some sort of numerical representation in a vector space is called “embedding.” Here, the word embedding is very simplistic and does not encode any sort of relationship between words, other than a sort of indexing. In order to represent each word in a way that efficiently carries some type of relationship of similarity, we have to move toward a dense vector representation, or a vector that is made up of mostly nonzero indices. While the processing of data happens token by token, it is each type that has a representation in a vector space. The vector spaces for the previous examples have three dimensions so that it is easy to imagine them in physical space, but the embeddings used in the projects described in this thesis all have upwards of 50 dimensions.

One simple way of moving from a sparse to a dense vector representation is through a co-occurrence count. Take, for example, the phrase “I am that I am.” This, if considered as our entire corpus, consists of five tokens and three types. We must first define the size of our context window, which is how many words before or after each token we will consider the “context” of that token. As an example, consider the word “that,” which occurs as the third word in this corpus. If the context window is one, then that includes one token before and one token after, meaning that the context window for “that” consists of “am” and “I.” If we expand that context window to two, then it would include "I," "am," "I," and "am." To create a co-occurrence vector representation, we define the size of our context window and count each time a token occurs in the context of another token.

	I	am	that
I	0	2	1
am	2	0	1
that	1	1	0

Figure 2. *Co-occurrence matrix for "I am that I am" with a context window of one.*

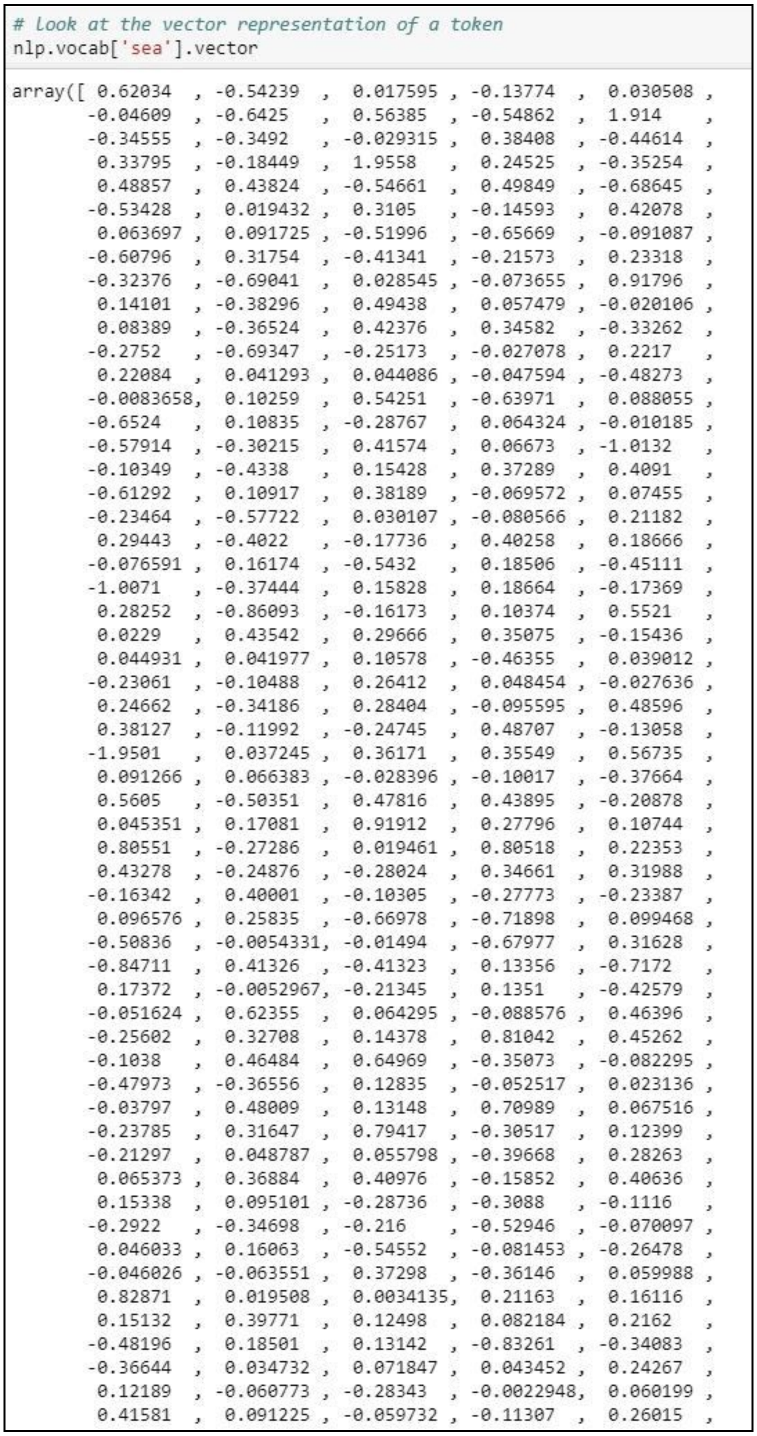
We can now go through the text one token at a time with a context window of one. In the first column for "I," we add one count of "am," because that is the only word before or after this token. Next, in the "am" column, we mark one for "I" and one for "that." Then, for the "that" column, we note that there is one co-occurence of "am" and one of "I." Then, because "I" is represented as a type, we consider the second "I" in the first column, and add another count of "am" and a count of "that." Lastly, we have another occurrence of the "am" type, and add a counter for the "I" that precedes it. This results in a vector representation that includes more information than the one-hot encoding. This changes the relationship of the representative vectors so that they will have some type of weighted relational similarity. This similarity can encode many different types of linguistic relationships. The co-occurrence vector is one way to heuristically encode semantic features, and is inspired by the distributional hypothesis in linguistics that "a word is characterized by the company it keeps."[16]

[16] J. R. Firth, "A Synopsis of Linguistic Theory 1930-1955,"iIn *Studies in Linguistic Analysis*, 1–32. Oxford, 1962.

Still, this small, isolated example does not really encode anything very meaningful. To do that, you must increase the amount of data that you analyze and embed, which will increase the number of dimensions beyond what we are capable of visualizing in physical space. Neural networks, or computational models inspired by the structure of the human brain that can automatically detect patterns in large amounts of data, can also be used to move from a sparse to a dense, continuous vector space and create word embeddings. We can use pre-trained word vectors, meaning that the embedding process is done already and included in some available software, to experiment with more meaningful vector representations. Pre-trained word vectors from GloVe (Global Vectors for Word Representation)[17] are available through spaCy, an open source library for natural language processing.[18] If we use the first ten chapters of *Moby Dick* as our corpus, we can use the GloVe word embeddings to see which words are most similar to other words. First, if we look at the word "sea," we can see what the exact vector representation is.

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "GloVe: Global Vectors for Word Representation," 2014. https://nlp.stanford.edu/projects/glove/.

[18] Explosion, "SpaCy · Industrial-Strength Natural Language Processing in Python." https://spacy.io/.



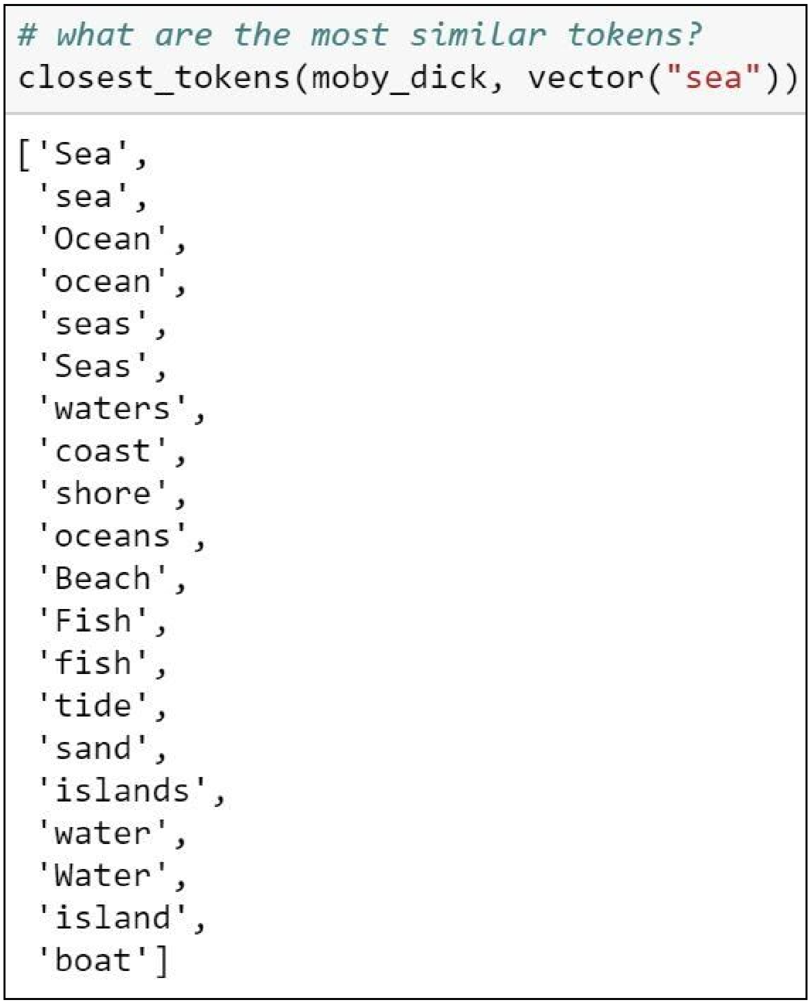
```
# Look at the vector representation of a token
nlp.vocab['sea'].vector
```

```
array([ 0.62034  , -0.54239  ,  0.017595 , -0.13774  ,  0.030508 ,
       -0.04609  , -0.6425   ,  0.56385  , -0.54862  ,  1.914    ,
       -0.34555  , -0.3492   , -0.029315 ,  0.38408  , -0.44614  ,
        0.33795  , -0.18449  ,  1.9558   ,  0.24525  , -0.35254  ,
        0.48857  ,  0.43824  , -0.54661  ,  0.49849  , -0.68645  ,
       -0.53428  ,  0.019432 ,  0.3105   , -0.14593  ,  0.42078  ,
        0.063697 ,  0.091725 , -0.51996  , -0.65669  , -0.091087 ,
       -0.60796  ,  0.31754  , -0.41341  , -0.21573  ,  0.23318  ,
       -0.32376  , -0.69041  ,  0.028545 , -0.073655 ,  0.91796  ,
        0.14101  , -0.38296  ,  0.49438  ,  0.057479 , -0.020106 ,
        0.08389  , -0.36524  ,  0.42376  ,  0.34582  , -0.33262  ,
       -0.2752   , -0.69347  , -0.25173  , -0.027078 ,  0.2217   ,
        0.22084  ,  0.041293 ,  0.044086 , -0.047594 , -0.48273  ,
       -0.0083658,  0.10259  ,  0.54251  , -0.63971  ,  0.088055 ,
       -0.6524   ,  0.10835  , -0.28767  ,  0.064324 , -0.010185 ,
       -0.57914  , -0.30215  ,  0.41574  ,  0.06673  , -1.0132   ,
       -0.10349  , -0.4338   ,  0.15428  ,  0.37289  ,  0.4091   ,
       -0.61292  ,  0.10917  ,  0.38189  , -0.069572 ,  0.07455  ,
       -0.23464  , -0.57722  ,  0.030107 , -0.080566 ,  0.21182  ,
        0.29443  , -0.4022   , -0.17736  ,  0.40258  ,  0.18666  ,
       -0.076591 ,  0.16174  , -0.5432   ,  0.18506  , -0.45111  ,
       -1.0071   , -0.37444  ,  0.15828  ,  0.18664  , -0.17369  ,
        0.28252  , -0.86093  , -0.16173  ,  0.10374  ,  0.5521   ,
        0.0229   ,  0.43542  ,  0.29666  ,  0.35075  , -0.15436  ,
        0.044931 ,  0.041977 ,  0.10578  , -0.46355  ,  0.039012 ,
       -0.23061  , -0.10488  ,  0.26412  ,  0.048454 , -0.027636 ,
        0.24662  , -0.34186  ,  0.28404  , -0.095595 ,  0.48596  ,
        0.38127  , -0.11992  , -0.24745  ,  0.48707  , -0.13058  ,
       -1.9501   ,  0.037245 ,  0.36171  ,  0.35549  ,  0.56735  ,
        0.091266 ,  0.066383 , -0.028396 , -0.10017  , -0.37664  ,
        0.5605   , -0.50351  ,  0.47816  ,  0.43895  , -0.20878  ,
        0.045351 ,  0.17081  ,  0.91912  ,  0.27796  ,  0.10744  ,
        0.80551  , -0.27286  ,  0.019461 ,  0.80518  ,  0.22353  ,
        0.43278  , -0.24876  , -0.28024  ,  0.34661  ,  0.31988  ,
       -0.16342  ,  0.40001  , -0.10305  , -0.27773  , -0.23387  ,
        0.096576 ,  0.25835  , -0.66978  , -0.71898  ,  0.099468 ,
       -0.50836  , -0.0054331, -0.01494  , -0.67977  ,  0.31628  ,
       -0.84711  ,  0.41326  , -0.41323  ,  0.13356  , -0.7172   ,
        0.17372  , -0.0052967, -0.21345  ,  0.1351   , -0.42579  ,
       -0.051624 ,  0.62355  ,  0.064295 , -0.088576 ,  0.46396  ,
       -0.25602  ,  0.32708  ,  0.14378  ,  0.81042  ,  0.45262  ,
       -0.1038   ,  0.46484  ,  0.64969  , -0.35073  , -0.082295 ,
       -0.47973  , -0.36556  ,  0.12835  , -0.052517 ,  0.023136 ,
       -0.03797  ,  0.48009  ,  0.13148  ,  0.70989  ,  0.067516 ,
       -0.23785  ,  0.31647  ,  0.79417  , -0.30517  ,  0.12399  ,
       -0.21297  ,  0.048787 ,  0.055798 , -0.39668  ,  0.28263  ,
        0.065373 ,  0.36884  ,  0.40976  , -0.15852  ,  0.40636  ,
        0.15338  ,  0.095101 , -0.28736  , -0.3088   , -0.1116   ,
       -0.2922   , -0.34698  , -0.216    , -0.52946  , -0.070097 ,
        0.046033 ,  0.16063  , -0.54552  , -0.081453 , -0.26478  ,
       -0.046026 , -0.063551 ,  0.37298  , -0.36146  ,  0.059988 ,
        0.82871  ,  0.019508 ,  0.0034135,  0.21163  ,  0.16116  ,
        0.15132  ,  0.39771  ,  0.12498  ,  0.082184 ,  0.2162   ,
       -0.48196  ,  0.18501  ,  0.13142  , -0.83261  , -0.34083  ,
       -0.36644  ,  0.034732 ,  0.071847 ,  0.043452 ,  0.24267  ,
        0.12189  , -0.060773 , -0.28343  , -0.0022948,  0.060199 ,
        0.41581  ,  0.091225 , -0.059732 , -0.11307  ,  0.26015  ,
```

Figure 3. *The vector representation of the token "sea" based on GloVe word embeddings in spaCy. This code is based on Allison Parrish's tutorial of word vectors.*[19]

[19] Allison Parrish, "Understanding Word Vectors: A Tutorial for 'Reading and Writing Electronic Text," a Class I Teach at ITP." GitHub. https://gist.github.com/aparrish/2f562e3737544cf29aaf1af30362f469.

This is much more dense than our previous examples, and has many more dimensions—300 to be exact. If we compare this vector to every other token in the first ten chapters of *Moby Dick*, we can see which tokens are most similar.



```
# what are the most similar tokens?
closest_tokens(moby_dick, vector("sea"))
```

```
['Sea',
 'sea',
 'Ocean',
 'ocean',
 'seas',
 'Seas',
 'waters',
 'coast',
 'shore',
 'oceans',
 'Beach',
 'Fish',
 'fish',
 'tide',
 'sand',
 'islands',
 'water',
 'Water',
 'island',
 'boat']
```

Figure 4. *The twenty closest tokens to "sea" in the first ten chapters of Moby Dick. This code is based on Allison Parrish's tutorial of word vectors.*[20]

All twenty of these tokens are related to the word "sea," but this semantic relationship is not hard-coded. The computer does not need to have explicit information about what the sea is, where it is usually located, or what types of things that you find in the ocean. Instead, this sort of

[20] Parrish, "Understanding Word Vectors."

information is encoded implicitly through the vector relationships.

In general, if machine learning algorithms are going to be applied to text, the textual or linguistic data must be supplied in some kind of numerical representation. Neural networks and other machine learning algorithms can be initialized with sparse vector representations that are iteratively or recursively changed in a meaningful way.[21] The computational process in which the weights and biases of a neural network are changed in order to automatically encode meaningful patterns in data is called training. There are different types of neural networks, however, and some lend themselves to the processing of sequential data like words or characters that occur in a sequence and make up a text. The recurrent neural network (RNN) is often used for processing sequential data. Recurrent neural networks take in input vectors and return output vectors with the definitive feature of changing the vector contents based not only on the input vector being looked at, but all past input vectors that it has seen. This ability to take into account past input vectors allows the RNN to have a better memory and maintain coherence for longer periods over a sequence.[22] The long-short-term-memory (LSTM) model is a type of RNN that builds upon this structure by further improving the ability to store past information.

The projects in this thesis feature the development of novel word embedding techniques, and the use of rule-based systems, LSTMs, and convolutional neural networks (CNNs), which are a different type of neural network frequently used for processing image data and handling classification tasks. The process of generating text using a trained neural network also frequently involves the initialization of a sequence, called a seed. This seed phrase or word or number is

[21] Matthew Honnibal, “Embed, Encode, Attend, Predict: The New Deep Learning Formula for State-of-the-Art NLP Models,” Explosion, November 10, 2016. https://explosion.ai/blog/deep-learning-formula-nlp.
[22] Andrej Karpathy, “The Unreasonable Effectiveness of Recurrent Neural Networks,” May 21, 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

used to start a process of generation that is specified by the programmer of the model. A seed is given, and then the next most likely character or word is chosen based on features learned by the trained neural network.

Vector space models have their roots in information retrieval with the vector representation of text documents, and were later expanded to other forms of data such as words.[23] Vector representations of data have long been a part of the infrastructure of our digital information systems and have increasing influence on the way that users of commercial networked platforms encounter their digital environments. From Spotify's personalized song recommendations[24] to Instagram's Explore feed,[25] to targeted ads and Amazon product recommendation services, the logic of machine learning is an integral part of many daily experiences.[26] This mundane reality, combined with sensationalized cultural conceptions of the slippery umbrella term of "AI," make the contemporary space of AI-generated literature an interesting area to critique, demystify, and complicate these processes and their influence on contemporary modes of being.

The process of embedding textual data in a vector space is also crucial to the inner workings of the machine learning and deep learning models that have dominated cultural conversations around AI and text-generation today, such as Google's BERT, which made waves in 2018 for its bidirectional approach of processing text, OpenAI's GPT-2, which is a language

[23] Gerard Salton, "Advanced Information-Retrieval Models: The Vector Space Model." In *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, (Addison-Wesley, 1989), 313–19

[24] Sophia Ciocca, "How Does Spotify Know You So Well?" Medium, October 10, 2017. https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe.

[25] James Vincent, "Instagram Explains How It Uses AI to Choose Content for Your Explore Tab," The Verge, November 25, 2019. https://www.theverge.com/2019/11/25/20977734/instagram-ai-algorithm-explore-tab-machine-learning-method.

[26] John D. Kelleher, *Deep Learning (The MIT Press Essential Knowledge Series)*, (Cambridge, MA: MIT Press, 2019).

model that uses another type of novel architecture called a transformer and caught the eye of the general public when the company that created it deemed it too dangerous to release because of its ability to so accurately mimic human-written texts.[27] The state of the art for natural language processing tasks is constantly changing. During the period between 2016 and 2018 when the main texts in this discussion were generated, the most popular AI text-generation models were LSTMs. Since then, the models have changed and continue to improve, but they are still enabled by the compatibility of vector representations with machine learning and deep learning models.

METHODOLOGY

In order to investigate emerging writerly practices, I focus on the craft of author/programmers who use and think about AI in their authorial practice and have published printed books of AI-generated literature.[28] I chose each of these authors and their associated works for the range of perspectives and creative practices that they provide. These are not the only author/programmers working with AI or the only books that fit my definition of AI-generated literature, but are a selection of creators with sustained practices that regularly engage with AI to achieve some sort of literary goal. The primary sources analyzed in this thesis are: 1) *Articulations* by Allison Parrish 2) *1 the Road* by Ross Goodwin, and 3) *ReRites - Raw Output / Responses* by David Jhave Johnston.

These works differ in a lot of ways. They are all published in different countries, albeit in three countries that have long histories of publishing computer-generated literature, and they all

[27] Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever, “Better Language Models and Their Implications,” OpenAI, February 14, 2019. https://openai.com/blog/better-language-models/.

[28] Nick Montfort and Judy Heflin, “Computer-Generated Literature, Author/Programmers, and Machine Voices,” in *The Cambridge Companion to Literature in the Digital Age* (Cambridge University Press: Forthcoming).

feature different genres of literary output. *Articulations* consists mainly of a 72-page prose poem and was published in the United States. *1 the Road* is a novelistic project that aims to automate the traditional American road trip, but was published in France. *ReRites* is a collection of lyric poetry, mostly in free verse, and was published in Canada. Each project focuses on a different aspect of machine writing and provides different solutions to the questions raised by AI-generated text, interfaces, and machine endurance.

The text-generation mechanisms behind these books are all enabled by vector representations of linguistic data and use machine learning techniques. They were all published in the last two years in the form of a printed codex book by small presses. There is no ebook option for any of these books. Each book is labeled explicitly as something that falls under the category of artificial intelligence. There is at least some code publically available to analyze for each book, and the code was written by an individual author/programmer. These books all incorporate some type of performance and creative subversion of industrial tools. They all provide contextualizing material to illustrate the human labor that went into their production in the form of introductions, essays, and sometimes "raw output" or additional literary work. After selecting these three primary sources, my process of analysis included:

- Analyzing available code and the "writing" process for each of the works, paying close attention to the material trajectory of textual data as it is created, cataloged, and computationally represented and manipulated, finally ending up on the printed page.
- A material and textual analysis of the books themselves, in order to inform ways of reading AI-generated literature and analyze AI-generated text in the form of a printed book.

- Reading paratextual elements, distributional press materials, and interpretations of these works from literary communities in the form of book reviews to further contextualize the work within the literary ecosystem.
- Personally questioning the authors about their craft, the way they conceptualize AI, and how and why they wanted to engage with AI in a literary way, both generally in relation to their broader practice and specifically for the book projects that I focus on in this discussion.

LITERATURE REVIEW

Many conceptual and theoretical frameworks have been proposed for thinking about and categorizing computer-generated literature. In his recent book *Electronic Literature*, e-lit scholar Scott Rettburg identifies threads of practice that can be understood as genres within the discipline, which is defined by the Electronic Literature Organization as "work with an important literary aspect that takes advantage of the capabilities and contexts provided by the stand-alone or networked computer."[29] Rettberg's five genres include: combinatory poetics, hypertext fiction, interactive fiction and other gamelike forms, kinetic and interactive poetry, and network writing.[30] Under this framework, the works focused on in this thesis fall under the category of combinatory poetics, in which Rettberg includes the subcategory of machine-learning-generated literature: big data poetics. This subgenre of big data poetics includes literary texts that are generated using contemporary approaches of artificial intelligence and machine learning that require large amounts of data. He notes that these systems are much more "black-boxed" than

[29] N. Katherine Hayles, "Electronic Literature: What Is It?" Electronic Literature Organization, January 2, 2007. https://eliterature.org/pad/elp.html.

[30] Scott Rettberg, *Electronic Literature* (Cambridge, UK: Polity Press, 2019).

other types of literary text-generation, and that “after training the corpus and configuring it, the developer plays little role in the process.”[31]

The broader category of AI-generated literature complicates and expands this subgenre. David Jhave Johnston’s earlier project that led to *ReRites* is Rettberg’s main example of big data poetry, which, he claims, raises the question of “a new role for the human poet: as editor for poetry entirely generated by machine intelligence.”[32] While this type of work has certainly spawned new editorial relationships and techniques, I argue that the developer of a literary text-generation system that uses machine learning engages in writerly practices that are not just editorial. Additionally, the part of a system that uses machine learning in some way is often framed by various authorial decisions that further inform the work. These decisions often consist of a computational solution to some sort of literary aspect of the work and necessitate computational definitions of literary categories. Additionally, I do not consider the defining characteristic of this kind of work to be the size of the dataset, but rather the utilization of machine learning algorithms that are enabled by the vector representation of linguistic data. AI-generated literature more broadly should encompass not only these types of work, but also computational literary works that are engaged with the social and computational histories of AI.

Rettberg importantly points out that the available machine learning models are largely the property of major corporations. This corporate relationship to large datasets, machine learning, and AI is a crucial aspect of both the reception and creation of this type of work. Social perceptions of AI are inherently linked to notions of labor and the very real fear of livelihoods being automated. State-of-the-art AI models and hardware are often entirely controlled and

[31] Rettberg, *Electronic Literature*, 53.
[32] Ibid., 53.

distributed by tech giants. Even if a creator of AI-generated literature chooses to develop their own system independent of megacorporations, they often do so in response to ideologies perpetuated by corporate software.

While Rettberg's genres are useful for situating the books I investigate within the history and practices of electronic literature more generally, they do not add much critical perspective to a close analysis of AI-generated books in print. The static material form of the printed book disqualifies the works that I am most interested in from traversing into the territory of hypertext fiction, interactive fiction, or kinetic poetry. Although these projects are not, "written for and published on the Internet," as is mandated by the last genre of network writing, they are all enabled by and involved with the internet in key ways. Under Rettberg's framework, I would classify *Articulations*, *1 the Road*, and *ReRites* under combinatorial poetics with a toe in network writing. As I will explain in more detail in Chapter 2, *Articulations* is the most traditionally combinatorial of the three works, as it arranges and combines lines of public domain poetry. However, as we look at the more computationally expensive systems used to generate *1 the Road* and *ReRites*, the combinatory mechanisms become so complex and extensive that the category begins to fall apart.

One of the most widely recognized antecedents to this type of computational text-generation is William Burroughs' cut-up method, which involved cutting a physical paper of text with scissors and mixing up the pieces to create something new. Burroughs writes that "All writing is in fact cut-ups. A collage of words read heard overheard. What else?"[33] It is true that neural networks and deep learning models combine words in some sense, but their vocabularies

[33] Nick Montfort, "The Cut-Up Method of Brion Gysin," in *The New Media Reader* (Cambridge, MA: MIT Press, 2003), 89–91.

are so extensive and the encoded understandings of language are so granular that the definitive quality of their work does not lie entirely in their combinatorial features.

Within the genre of combinatorial poetics, computational poet and professor of digital media Nick Montfort has demonstrated that computational writing has independent and identifiable techniques. He proposed a preliminary typology of four characterizing properties of this kind of work: exhaustive vs. sampled, static vs. dynamic source texts, simple vs. elaborate systems, and one medium vs. multimedia.[34] *Articulations*, *1 the Road*, and *ReRites* are all examples of sampled texts, meaning that rather than generating every possible combination that could be output by a literary text-generation system, the program selects or generates only some of what it is capable of creating. The exhaustive potential of machine learning combined with the material limitations of the printed book make this almost a necessary quality of the types of projects I am looking at. In this case, each book has static source texts. Although the source texts change throughout the many books in the broader *ReRites* project, each volume has its own static source text. Each process for generating the text is quite elaborate, requiring relatively large amounts of data and code. They are all one medium: text. Although *1 the Road* features multimodal ways of obtaining textual data, everything ends up as text on the printed page, including GPS data, audio recordings, time data, and image data. Even the image data is run through a program that translates images into ASCII artwork, resulting in images made up of text. This typology is only preliminary, and through the close analysis of three works that all fall into the same categories of *sampled*, *static source texts*, *elaborate systems*, and *one medium*, it becomes apparent that there are a range of differences in techniques and practices even among

[34] Nick Montfort, "Conceptual Computing and Digital Writing," in *Postscript: Writing After Conceptual Art*, edited by Andrea Andersson, (Toronto: University of Toronto Press, 2018), 197–210.

these similar texts. As more computer-generated work is created, a more elaborate typology can and should be developed. This thesis provides a framework for thinking about creative text-generation techniques that are used for *Articulations*, *1 the Road*, and *ReRites* in Chapter 3.

Media studies scholar Lori Emerson writes that networked technologies have already transformed the role of the writer and the nature of writing itself. The automatic surveillance and manipulation of text as it is entered into the network creates a feedback loop between reading and writing that she calls *readingwriting*.[35] She argues that the twenty-first century brought with it a renewed interest in the algorithms that are reading, writing, and reading our writing, in a dialogue that occurs through the network. This is especially relevant to the close corporate oversight of computer-generated work that utilizes machine learning and artificial intelligence. Just as Emerson's examples of *readingwriting* simultaneously use Google and critique Google, each of the works discussed in this thesis has a unique relationship with networks and corporations. Many of these works are also supported in part by the corporations that they are critiquing.

Electronic literature scholar Leonardo Flores has proposed a continuum of distance to describe the range of writerly practices in computer-generated literature. His framework for "Distant Writing," as opposed to what he calls "Close Writing," where writers are manually responsible for the creation of a text, is roughly divided by the distinctions of short, medium, and long distances.[36] Short distances include works where authors have full control over the "engine" and dataset used to generate a text. The author/programmer writes the dataset and all of the code

[35] Lori Emerson, *Reading Writing Interfaces: From the Digital to the Bookbound* (Minneapolis: University of Minnesota Press, 2014), 185.
[36] Leonardo Flores, "Distant Writing," YouTube, MITH Digital Dialogues, March 10, 2020. https://youtu.be/It2seaB5x-U.

for the system. Projects where the data or source texts are chosen rather than created and subsequently altered are categorized at a medium distance, while works that use machine learning, neural networks, or artificial intelligence are labelled long distance. These techniques are labeled as the most-distant category because, "writers exert minimal or no control."[37] While it is true that certain linguistic and therefore literary aspects of a computer-generated work are automatically encoded if an author/programmer chooses to utilize some type of machine learning technique, it is not the case that writers exert minimal or no control. Through looking closely at three contemporary works of AI-generated literature, it is evident that a range of techniques are used for various types of authorial control.

This language of distance is similar to the analytical continuum from authorship to generatorship proposed by Leah Henrickson, wherein computer-generated texts can be can placed on a continuum between the machine and the human in order to complicate the binary options of attributing a text to either a human or computer author.[38] This continuum might be more useful for commercial applications of text-generation or instances of authors collaborating with systems that have been developed by teams of programmers with seamless user interaction in mind, such as the "Synthetic writing" experiment from the Antwerp Center for Digital Humanities and Literary Criticism.[39] It is not as useful for the analysis of texts that have been created by individual author/programmers who have intentionally created a work of literature using machine learning techniques.

[37] Flores, "Distant Writing."

[38] Leah Henrickson, "Towards a New Sociology of the Text: the Hermeneutics of Algorithmic Authorship," (PhD diss., Loughborough University, 2019).

[39] Enrique Manjavacas, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont, "Synthetic Literature: Writing Science Fiction in a Co-Creative Process," *Proceedings of the Workshop on Computational Creativity in Natural Language Generation*, 2017. https://doi.org/10.18653/v1/w17-3904.

Still, the language of distance is useful in some ways. Each author/programmer in this thesis emphasizes attention and closeness to a specific intervention point in the development of a machine-learning-powered system. These intervention points are most easily described by the continuum of human-computer interaction proposed in Espen Aarseth's 1997 typology of "Cyborg Literature."[40] Aarseth points to the necessity of human or authorial labor in any kind of creative computational text-generation system, and describes three main positions of human-machine collaboration: preprocessing, in which a system is developed by an author; coprocessing, in which the author interacts with a system to generate text; and postprocessing, in which the author engages in some sort of editorial process after text is generated. While computer-generated literature entails at least some preprocessing, there are many ways that a work can include some, or all three, of these "cyborg-author combinations."[41] Among the three works of AI-generated literature featured in this thesis, each focuses mainly on one of these three types of processing, each highlighting a different kind of closeness to the machine. I argue that AI-generated literature that uses machine learning to generate text does not obsolesce the human from textual production as some scholars have argued,[42] but rather encompasses a range of emerging and sometimes divergent authorial practices.

In an effort to move literary criticism away from the limits of intentional readings, Roland Barthes claimed that, "To give an Author to a text is to impose upon that text a stop clause, to furnish it with a final signification, to close the writing."[43] This thesis is not an effort to

[40] Espen J. Aarseth, *Cybertext: Perspectives on Ergodic Literature* (Baltimore, MD: Johns Hopkins University Press, 1997), 134.
[41] Aarseth, *Cybertext*, 135.
[42] Henrickson, *Towards a New Sociology of the Text*, 181.
[43] Roland Barthes, "The Death of the Author," in *Image / Music / Text*, translated by Stephen Heath, (London: Fontana, 1977), 142–47.

focus only on authorial intention and preferred interpretations of computer-generated texts, but rather a way to clearly lay out and label the authorial labor that goes into AI-generated literature in order to open up the text and inform more nuanced readings. Machine learning and the "black-boxing" of contemporary AI technologies seems to cast a shadow of obfuscation on any text that claims to be AI-generated. I hope to bring some light to the very interesting works that have been created using machine learning and to provide some initial ways to think about this type of writing as AI resources become more advanced and widely available for writers to use.

I also focus on the relationship between these text-generation systems and the printed book and argue that the material codex book is also a key area of analysis for this kind of work. Although the Electronic Literature Organization specifically worded their definition of e-lit so that it could include computer-generated poetry that was printed in the form of a book,[44] this type of work seems to be the exception rather than the rule. Both Katherine Hayles and Scott Rettberg have noted that electronic literature is not usually printed in the form of a book.[45, 46] Despite being somewhat of an outlier within the broader discipline, there is a rich history of computer-generated books that dates all the way back to 1964. Katherine Hayles has proposed a framework of "intermediation" that can be used to think about the printed book in the context of electronic literature in a way that is nuanced enough to avoid the pitfalls of reductively importing theoretical assumptions developed in the context of print to the analysis of electronic media.[47] The idea of "intermediation" is similar to the feedback loops of Lori Emerson's readingwriting, but focuses on what Hayles calls "dynamic heterarchies" in which complexity emerges through

[44] Hayles, "Electronic Literature: What Is It?"
[45] Hayles ibid.
[46] Rettberg, *Electronic Literature*, 5.
[47] N. Katherine Hayles, *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005) 7.

recursive interactions between mediated physical, psychological, economic, and social formations.[48] Computer-generated literature more broadly interacts with the printed book in significant ways, but is also one of many modes of analysis for literary AI-generated text in the form of a printed book, which will be discussed further in Chapter 4.

[48] N. Katherine Hayles, "Intermediation: The Pursuit of a Vision," *New Literary History* 38, no. 1, (Winter 2007), 100.

2. PRIMARY SOURCES AND SYSTEMS

In his lecture, Calvino concludes that "the struggle of literature is in fact a struggle to escape from the confines of language," and that the computer-generated writing undertaken by the avant-garde is another attempt to do so.[49] Along the lines of the avant-garde tradition, Allison Parrish says that through her work she is "mostly interested in breaking things, and computation just happens to be a really helpful hammer for doing that."[50] She sees computation as "the latest in a line of tools," and is specifically drawn to machine learning for its powerful ability to create the sorts of juxtapositions that she is interested in.[51] Katherine Hayles argues in favor of computation as an expressive tool or medium, saying that "compared to a hammer [...] a computer has much more flexibility, interactivity, and cognitive power."[52] Ross Goodwin is also interested in "breaking literature," but through the development of AI-powered interfaces that make up what he calls "narrated reality," in which everything can be experienced as computationally-mediated text. David Jhave Johnston interacts with AI-generated literature as an act of collaboration and treats the generated text as a medium that he "carves." For Johnston, his mind acts like more of a hammer than the text-generation system: "It is 6 am. It's silent. The internet is off. Mind is hammer. I carve."[53] AI can be seen as a tool, an interface, or a medium, and the way that author/programmers conceptualize AI frames the goals of their work and guides the literary and creative potential of the technologies.

This chapter gives an overview of the creative practices and text-generation techniques

[49] Calvino, *Cybernetics and Ghosts*, 18.
[50] Allison Parrish (Author of *Articulations*), in discussion with the author, February 20, 2020. See Appendix A.
[51] Parrish, Appendix A.
[52] Hayles, *Intermediation*, 102.
[53] David Jhave Johnston, "RERITES : Machine Learning Poetry Edited by a Human." Glia, n.d. http://glia.ca/rerites/.

used to write *Articulations*, *1 the Road*, and *ReRites*, and brings attention to the writerly decisions made at different points in the text-generation process. I am dividing the writerly process of these works of AI-generated literature into the categories of datification, vectorization, generation, and pagination. Datification is the stage at which a writer either creates, compiles, or selects the data that is necessary to train some sort of machine learning program. Vectorization is the process by which this data is translated into a vector representation and used in a machine learning program. The process of generation is the actual creation of a textual output through running some sort of program that interacts with a machine learning model in some way. The final step, which is necessary for the eventual act of printing the output is pagination, which explores the editorial processes involved post-generation and the final formatting of text for the printed book.

ARTICULATIONS

Allison Parrish's practice focuses on the intersection of computation and language. In recent years, her work has been focused on the exploration of vector representations of linguistic data and new forms of poetic operations afforded by vectorization, which is the operation that has to happen in order for data to be legible to machine learning processes. Allison Parrish is a computer programmer who develops her own code and computational models for poetic exploration, unlike practices such as Google autocomplete poetry[54] that use pre-existing tools and don't require programming knowledge. In *Reading Writing Interfaces*, Lori Emerson concludes that the primary difference between computer-generated poetry and poetry that is the

[54] Catherine Armitage, "Google Autocomplete Is Being Used to Create Poetry," The Sydney Morning Herald, February 2, 2014.
https://www.smh.com.au/opinion/google-autocomplete-is-being-used-to-create-poetry-20140201-31tp4.html.

result of a search engine query "is that in the former the poet or programmer creates the dataset, as well as the grammatical and syntactical parameters, from which the computer creates a poem and that in the latter the poet initiates a search in order to obtain a data set from which to draw."[55] The creation of a dataset and any programming of linguistic relationships or parameters is part of the writerly process of AI-generated literature.

Articulations is the third release in the Using Electricity series of computer-generated books published by Counterpath Press. The book consists of two parts: Part I, Tongues, which is a prose poem that takes up 72 of the book's 105 pages, and Part II, which is a shorter collection of poems of lineated verse titled Trees. Tongues is the output of a random walk through the vector space of over three million lines of public domain poetry collected from Project Gutenberg, resulting in a phonetic gradient of clauses spanning the history of English language verse in a noisy timeline that begins with *The Suppressed Poems* of Alfred, Lord Tennyson and ends with *Rivers to the Sea* by Sara Teasdale. At each step in the random walk, a new line of poetry is added to the poem. Part II, Trees, uses a similar technique of exploring distances in a vector space, but in a different form and using different features. In Part II, each poem is a random walk through a vector space of syntactic features as opposed to the phonetic features of Part I. This technique of accumulating fragments of different source texts and poetic voices, mediated by vectorial distance, is clearly part of the intention of the author. However, the authorial intent opens the writing beyond, "the message of the AuthorGod," and into, "a multi-dimensional space in which a variety of writings, none of them original, blend and clash." [56] *Articulations* is truly a "tissue of quotations drawn from the innumerable centres of culture."[57]

[55] Emerson, *Reading Writing Interfaces*, 178.
[56] Barthes, *The Death of the Author*, 146.
[57] Ibid.

Barthes was describing human-authored texts in this way, but this is also an uncannily accurate description of *Articulations* as it compiles quotations from writers throughout history and organizes them phonetically, placing emphasis on how they are spoken in the human body. Through an analysis of the writing process of *Articulations*, we can explore the intermediation of speech, writing, and code, as it relates to conceptions of artificial intelligence, computer-generated literature, and the printed book today.

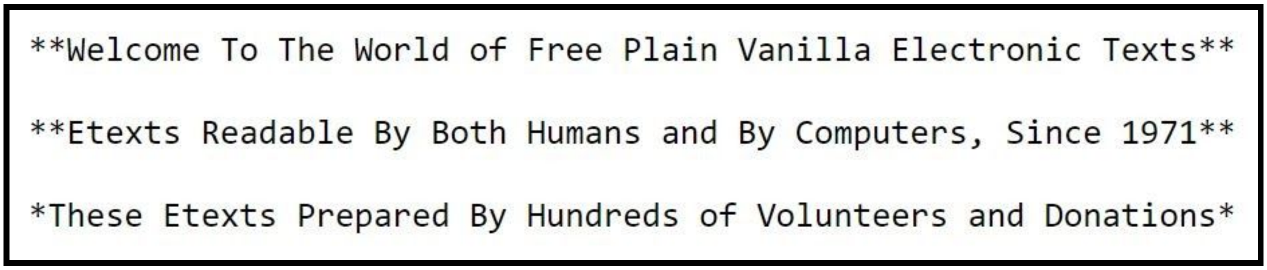
Datafication

The first step in both Parrish's writing and our understanding of *Articulations* will be to examine the process of how the etexts of Project Gutenberg went from being tens of thousands of plain text files hosted online to a neatly gzipped json file of 3,085,117 lines of poetry. The starting point for most computer-generated writing projects involves the creation and/or selection of datasets. Most machine learning projects require datasets that are much more extensive than smaller-scale combinatorial programs that, for example, exhaustively permute a limited vocabulary or grammar. This entails more ambitious efforts for data collection and curation, in which many poetic decisions will inevitably be made. The choice of data, especially when the dataset becomes more involved than just selecting words and begins to venture into the appropriation of language that is written by others, also has severe implications for the meaning of the work. The philosophy of Project Gutenberg as an initiative, the textuality of its electronic document "media translations,"[58] and Parrish's computational decisions are all part of the reading and writing process of *Articulations*.

Project Gutenberg is a free and open-source archive of literature in the public domain that

[58] N. Katherine Hayles, "Translating Media: Why We Should Rethink Textuality," *The Yale Journal of Criticism* 16, no. 2 (2003), 263–290.

is 50 years in the making. The project began in 1971 when founder Michael Hart typed up the U.S. Declaration of Independence.[59] This was soon followed by the Bill of Rights, the U.S. Constitution, the Bible (book by book), and the entire works of Shakespeare (a play at a time). Once these crucial founding texts were typed up, formatted, and uploaded, the selection of texts fell into the three categories of "light literature," "heavy literature," and "reference." The goal was to distribute electronic texts, or "etexts," so that they cost so little that they were practically free, and to keep them in such a simple format that both machines and humans could read them easily. Over the years, the Project Gutenberg team set the lofty goal of uploading 1,000,000 books by 2015. While they didn't hit that goal, as of February 2020, the site has over 60,000 free ebooks available with entries in over 60 languages.



```
**Welcome To The World of Free Plain Vanilla Electronic Texts**

**Etexts Readable By Both Humans and By Computers, Since 1971**

*These Etexts Prepared By Hundreds of Volunteers and Donations*
```

Figure 5. *Excerpt from the Project Gutenberg Etext of The Declaration of Independence*

In *From Gutenberg to Google: Electronic Representations of Literary Texts*, textual studies scholar Peter Shillingberg laments the scholarly lossiness of the printed book as it is transformed into a Project Gutenberg etext, suggesting that the "plain vanilla" format is inaccurate—specifically, lacking in the recording of bibliographic features and changes made to the text by way of editing, transcribing, or scanning.[60] He notes that Project Gutenberg's lack of

[59] Michael Hart, "The History and Philosophy of Project Gutenberg," Project Gutenberg, August 1992. https://www.gutenberg.org/wiki/Gutenberg:The_History_and_Philosophy_of_Project_Gutenberg_by_Michael_Hart.
[60] Peter Shillingsburg, *From Gutenberg to Google: Electronic Representations of Literary Texts* (Cambridge University Press, 2006), 21.

thoughtful archival methods will result in “Noisy texts, without any doubt. Misleading texts, very likely. Texts useless for scholarly purposes, of course.”[61] However, Project Gutenberg did not have scholars in mind when attempting to disrupt traditional notions of publishing and distribution, at least in the first 30 years of its production. The audience for Project Gutenberg was “99% of the general public.”[62] In Project Gutenberg founder Michael Hart’s updated 1992 manifesto, *The History and Philosophy of Project Gutenberg*, he states that “we do not write for the reader who cares whether a certain phrase in Shakespeare has a ‘:’ or a ‘;’ between its clauses. We put our sights on a goal to release etexts that are 99.9% accurate in the eyes of the general reader.”[63]

It is undeniable that one cannot conduct the same bibliographic or scholarly investigation on a printed work and a scanned etext version of it. As Katherine Hayles notes, “no print book can be completely encoded into digital media.”[64] Even a scan of a book loses material information that is characteristic of the original object, and the Project Gutenberg etexts even moreso. This calls for a focus on processes and driving interests behind these sorts of media translations. Although each Project Gutenberg entry does not include thorough documentation of bibliographic and editorial methods, they do include the particular Project Gutenberg legal license that distinguishes how “free” the etext is, as well as the “production” credits—the name(s) of the volunteers who typed in or scanned and edited the text character by character into a plain vanilla ASCII file. Because of the voluntary nature of the work, the selection of titles was not methodical in the way of scholarly merit, but by way of encouraging volunteers to want to do

[61] Shillingsburg, *From Gutenberg to Google*, 22.
[62] Hart, *History and Philosophy of Project Gutenberg*.
[63] Ibid.
[64] Hayles, *Translating Media*, 270.

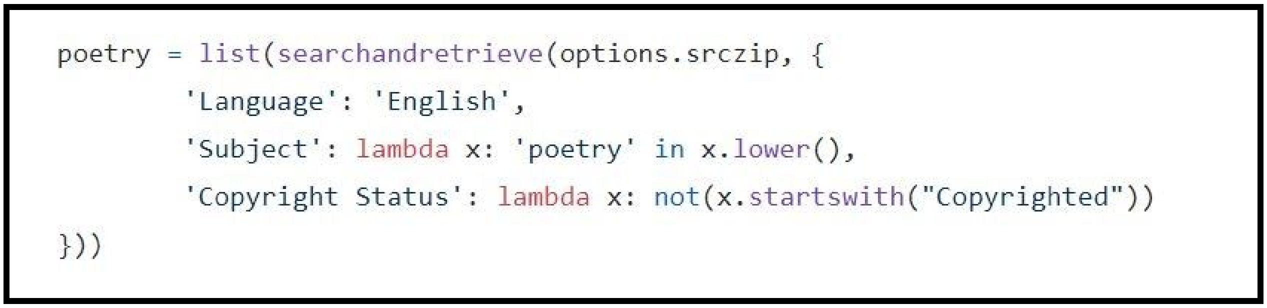
the work. Volunteers were encouraged to choose books in the public domain that they enjoyed and wanted to work on.[65] The Project Gutenberg database is a noisy and imprecise labor of love, reflecting a particular philosophy toward information management and freedoms that began with the rise of personal computing in the 1970s.

When Parrish was teaching a class in the creative writing department at Fordham University, she wanted to do an exercise related to Google autocomplete poetry, but didn't want her students to be swayed or confined by the values of Google Search. In order to facilitate this workshop without having to use Google, she created a poetry autocomplete interface for her students to use based on lines of poetry from Project Gutenberg. This involved creating two datasets. The first, Gutenberg, Dammit,[66] is a corpus of every plaintext file in Project Gutenberg up until June 2016. The second, A Gutenberg Poetry Corpus,[67] is a dataset of over three million lines of poetry made out of Gutenberg, Dammit. The Gutenberg, Dammit corpus is organized by way of metadata as classified by Project Gutenberg volunteers. Parrish automatically identified and extracted plaintext English language files from Project Gutenberg and organized them by way of existing metadata, as well as a few added categories such as the character set, and path to the file within Gutenberg, Dammit. The selection of texts as well as their "Language," "Subject," and "Copyright Status" classification within the Project Gutenberg organization scheme all provided the structure for what then became A Gutenberg Poetry Corpus.

[65] Hart, *History and Philosophy of Project Gutenberg*.

[66] Allison Parrish, "Gutenberg, Dammit," GitHub, August 2018. https://github.com/aparrish/gutenberg-dammit.

[67] Allison Parrish, "A Gutenberg Poetry Corpus," GitHub, August 2018. https://github.com/aparrish/gutenberg-poetry-corpus.

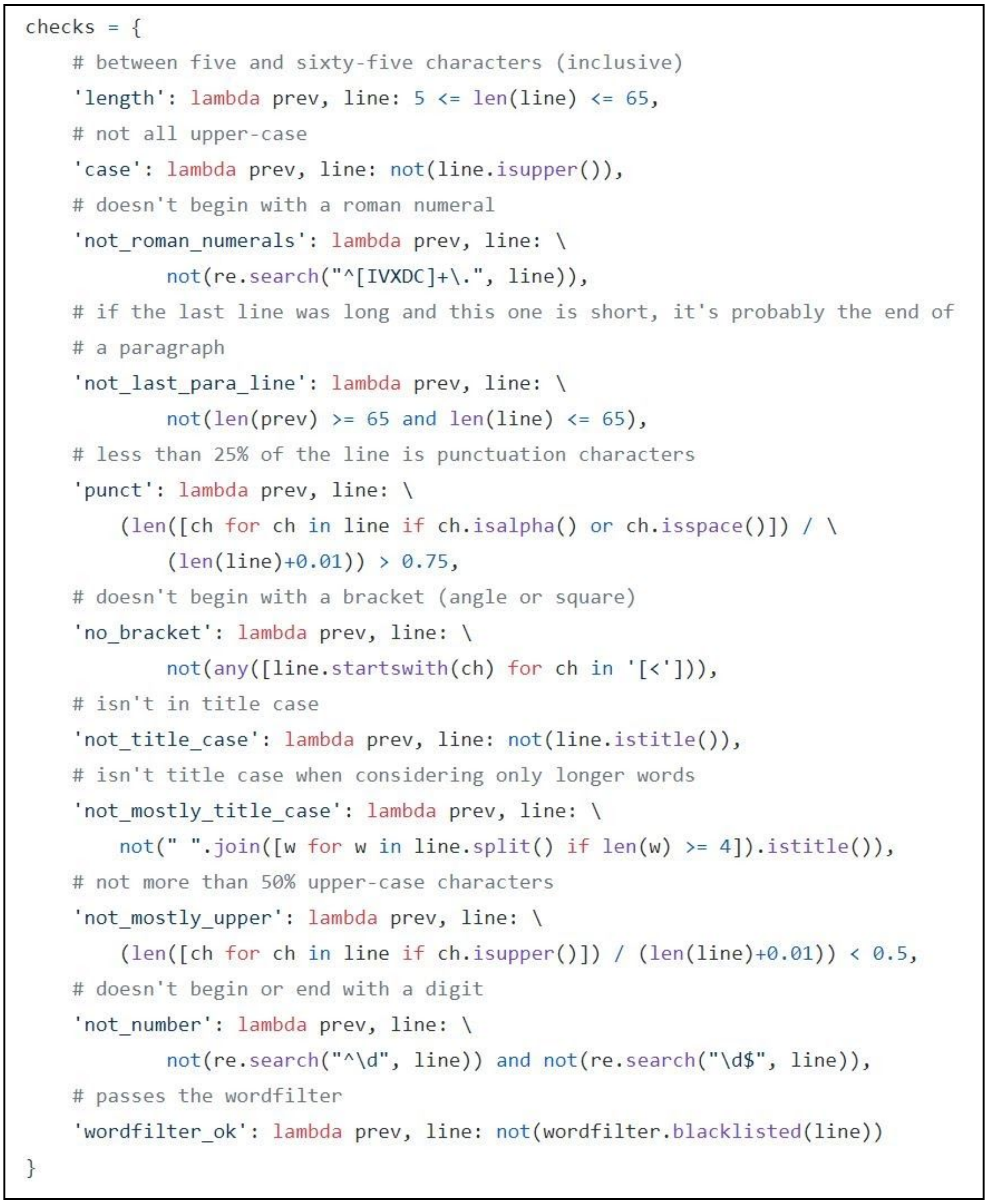


```
poetry = list(searchandretrieve(options.srczip, {
        'Language': 'English',
        'Subject': lambda x: 'poetry' in x.lower(),
        'Copyright Status': lambda x: not(x.startswith("Copyrighted"))
}))
```

***Figure 6**. Definition of a poem in A Gutenberg Poetry Corpus*

If the word “poetry” was anywhere in the “Subject” metadata tag for a given Project Gutenberg etext, it became a key building block of A Gutenberg Poetry Corpus. After selecting all of the works of poetry, Parrish developed a rough heuristic for defining a “line” of poetry as a unit. Parrish is interested in the line as a unit and describes a “line of poetry being analogous to a text file line being analogous to a database record,”[68] in a history that spans the range of Mesopotamian business records through contemporary computing cultures. She wanted to create a sort of container that she could randomly pull lines from in order to facilitate creative poetic experiments.

[68] Parrish, Appendix A.



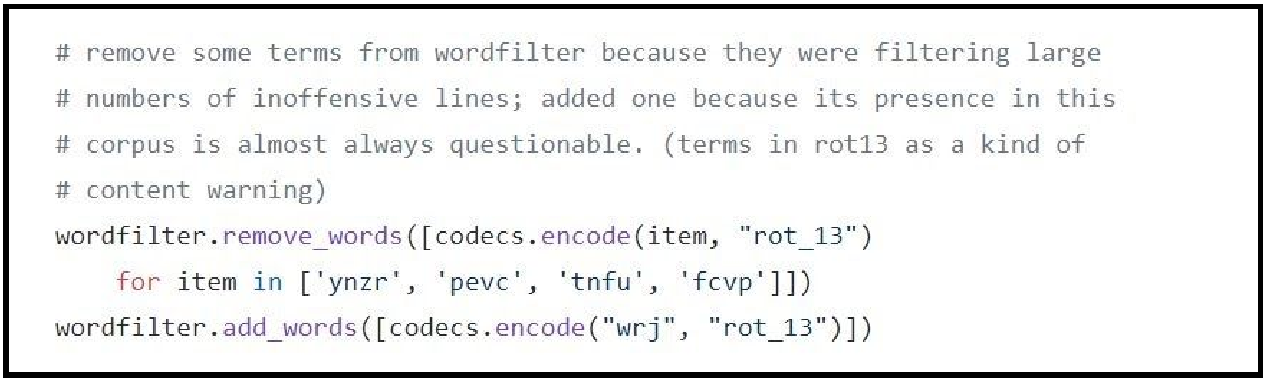
```
checks = {
    # between five and sixty-five characters (inclusive)
    'length': lambda prev, line: 5 <= len(line) <= 65,
    # not all upper-case
    'case': lambda prev, line: not(line.isupper()),
    # doesn't begin with a roman numeral
    'not_roman_numerals': lambda prev, line: \
            not(re.search("^[IVXDC]+\.", line)),
    # if the last line was long and this one is short, it's probably the end of
    # a paragraph
    'not_last_para_line': lambda prev, line: \
            not(len(prev) >= 65 and len(line) <= 65),
    # less than 25% of the line is punctuation characters
    'punct': lambda prev, line: \
        (len([ch for ch in line if ch.isalpha() or ch.isspace()]) / \
            (len(line)+0.01)) > 0.75,
    # doesn't begin with a bracket (angle or square)
    'no_bracket': lambda prev, line: \
            not(any([line.startswith(ch) for ch in '[<'])),
    # isn't in title case
    'not_title_case': lambda prev, line: not(line.istitle()),
    # isn't title case when considering only longer words
    'not_mostly_title_case': lambda prev, line: \
        not(" ".join([w for w in line.split() if len(w) >= 4]).istitle()),
    # not more than 50% upper-case characters
    'not_mostly_upper': lambda prev, line: \
        (len([ch for ch in line if ch.isupper()]) / (len(line)+0.01)) < 0.5,
    # doesn't begin or end with a digit
    'not_number': lambda prev, line: \
            not(re.search("^\d", line)) and not(re.search("\d$", line)),
    # passes the wordfilter
    'wordfilter_ok': lambda prev, line: not(wordfilter.blacklisted(line))
}
```

Figure 7. *Definition of a line in A Gutenberg Poetry Corpus*

A line of poetry, as defined by the Gutenberg Poetry Corpus, is a sequence of characters that is between five and sixty-five characters in length, is not all uppercase, does not begin with a roman numeral, is preceded by a line of similar length, is less than 25% punctuation, does not begin with angled or squared brackets, is not in title case, does not begin or end with a digit, and lastly, passes the wordfilter of blacklisted words as specified by Darius Kazemi's wordfilter library. This iterative list of constraints is a very computational definition of a line that effectively estimates and filters out character sequences that would be productive for computational creativity projects. It also points to the material reality of the Gutenberg etexts as they are created and maintained to be read by both humans and machines. Because the project started in the 1970s, the first Gutenberg etexts were produced in all capital letters because computers at the time did not use lower case. Despite being "plain vanilla" etexts, these types of stylistic decisions were made by the constraints of certain platforms and modes of textual inscription. The decision to later re-inscribe the etexts with case sensitivity was made by Project Gutenberg editors, and is an interesting counterpoint to one of Parrish's definitions of a line being, "isn't title case."

These heuristic methods for determining a line of poetry also critique and comment on the precariousness and subjectivity of software systems. Parrish says that a main focus of her practice is to, "create language that strongly suggests things."[69] This type of suggestion can be read and experienced at the surface level of the printed text of *Articulations*, but also in her programming and heuristic approaches at various points in the development of her code. A similar heuristic method was used to insert the paragraph breaks and indentations in Part I, Tongues.

[69] Ibid.



```
# remove some terms from wordfilter because they were filtering large
# numbers of inoffensive lines; added one because its presence in this
# corpus is almost always questionable. (terms in rot13 as a kind of
# content warning)
wordfilter.remove_words([codecs.encode(item, "rot_13")
    for item in ['ynzr', 'pevc', 'tnfu', 'fcvp']])
wordfilter.add_words([codecs.encode("wrj", "rot_13")])
```

Figure 8. *Wordfilter encryption*

One interesting poetic decision in Parish's process is the wordfilter and her modifications to it. The badwords.json file in the original wordfilter program is an alphabetized display of all of the most offensive words in the English language, that are presented in isolation without obfuscation. Parrish, in her decision to edit the badwords array, did so by way of a simple encryption cypher called ROT13, which stands for "rotate by 13 places," in order to represent offensive words in a way that the computer can understand but is obfuscated for the human programmer as a "kind of content warning." This filtering of language is also reflective of the responsibility that Parrish feels for the outcome of whatever programs she develops and runs, which is a philosophy that had significant implications for the editing, production, and printing processes of the book.

The dataset for *Articulations*, A Gutenberg Poetry Corpus, is the result of 50 years of volunteer labor, framed by philosophies of freedom of information within digital cultures, and reflective of a certain DIY ethos. Gutenberg, Dammit took advantage of the plaintext format emphasized by Project Gutenberg, and A Gutenberg Poetry Corpus utilized the existing classification work done by volunteers to gather a dataset that would provide the backbone for Parrish's students to experiment with autocomplete poetry outside the search monopoly of

Google, and lay the ground for further poetic experimentation.

Vectorization

For creating the learned latent space of *Articulations*, Allison Parrish created a novel method for phoneme vectorization. Using ARPABET phoneme transcriptions that map phonemes to their corresponding manner and place of articulation in the body, Parish was able to create a vector space that encoded relationships between articulatory features. This enabled her to then embed the lines of poetry from A Gutenberg Poetry Corpus within a phonetic vector space. Each line of poetry is placed quantitatively to a point in space that is a certain distance from all the other lines. This measure of distance encodes a relationship of phonetic similarity, meaning that lines that sound the same are closer together and lines that sound very different are further apart. The main poetic operations of vectorization that are of interest to Parrish are the unique abilities afforded by linguistic relationships that are encoded in distance. Once you vectorize words, or sentences, or whatever data you are working with, it is possible to do things like find the midpoint between two items. Through these types of relationships, Parrish explores additional experiments such as averaging the sound of a sentence, sound analogies, and sound tinting.[70] Parrish says that "these are new poetic operations that we haven't had access to before," and that "from a poetic standpoint, you wouldn't be able to do unless you had some kind of mathematical representation of words."[71]

While Part I, Tongues, focuses on these articulatory features, Part II acts as a counterpoint to contextualize and clarify the methods used in Part I. Part II, Trees, uses the same

[70] Allison Parrish, "Poetic Sound Similarity Vectors Using Phonetic Features," *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, September 19, 2017.
[71] Parrish, Appendix A.

means of determining similarity, but in a different form and using a different set of features, which means that the vector representations and resulting vector spaces are different for Parts I and II. Trees is based on syntactic features as opposed to phonetic features, meaning that the relationship between points in the vector space is determined by the similarity of the structure of a line and its component parts of speech. These structural elements are easier to identify in the shorter verses of Trees, and are provided as a companion element to the main phonetic project.[72]

Generation

This method of embedding stretches of text in a continuous vector space according to phonetic similarity is what opens up the possibility for the aesthetic gradient of sound that is created in *Articulations*. Equally as important as the poetic decisions that go into developing a dataset and a language model is the computational method of generation, in this case, how to journey through the resulting vector space and ultimately generate a text. Parrish sees the idea of displaying one thing after another that is slightly different from the previous one as the primary poetic operation of machine learning.[73] The method of generation or traversal in *Articulations* is an approximation of a random walk through the vector space. A random walk is the process by which a randomly moving object wanders away from a starting point one step at a time. This algorithm has been culturally significant in various domains ranging from theories about stock market investment strategies[74] to path planning algorithms for early roombas.[75]

The implementation of the random walk used for *Articulations* is again more of a suggestion or heuristically-motivated random walk rather than a formal mathematical

[72] Ibid.

[73] Ibid.

[74] Burton Malkiel, *A Random Walk Down Wall Street* (Norton, 1973).

[75] Simon Garfinkel, "IRobot Roomba," MIT Technology Review, October 9, 2002. https://www.technologyreview.com/2002/10/09/234680/irobot-roomba/.

instantiation. The algorithm that Parrish wrote picks a random vector from the vector space to begin, and then, keeping track of lines of poetry it has already seen, picks the closest line of poetry in the phonetic vector space. It then does this in a loop until the total number of steps as specified by Parrish is complete. After running this random walk, it became obvious that the algorithm would not be able to provide the sort of wide-ranging gradient she was looking for because it would get stuck in a nearby cluster and wouldn't be able to escape until it exhausted all of the closest lines. In order to introduce additional randomness into the algorithm, a velocity vector of a small uniform random number was added at each step, and that addition of slight variation in direction was enough to pop the algorithm out of the clusters and create more diversity in the output. Imagine a roomba floating through a phonetic vector space, vacuuming up lines written by poets throughout the history of Western literature and gluing them together.

Despite not being programmed for semantic properties, the phonetic similarity does result in an emergent type of semantic similarity, with clusters of topics that poets have been writing about across the centuries emerging. When the roomba turns a particular corner of ideas, you can tell, with sections on goodbying, goodnighting, thinking, and dying, and a lap around "happy,"[76] which eventually blends into "ha-ha-ha,"[77] and "he-he-he."[78] The random walk wasn't Parrish's only choice of method for generation, but the random walk served to create a longform chain made up of subtle juxtapositions and was the best way to represent what Parrish is interested in exploring through machine intelligence.

Articulations is not a narrative in the traditional sense of having a coherent story over the course of the text, but it is still a narrative of storts. If narrative is defined as a representation of

[76] Parrish, *Articulations*, 26.
[77] Ibid., 23.
[78] Ibid.

events in a sequence, then *Articulation*s is indeed a narrative made up of the approximation of a random walk that occurs in a sequence of steps during the traversal of a vector space.[79] Most readers will also begin to identify certain patterns and arcs of story while reading the book, especially when the text is considered out of context from the specifics of its computational modes of generation. Parrish, when doing readings of the book, specifically chooses passages that feel like they have a sort of beginning and end. She chose her go-to passage for readings because of the interesting phonetic transitions between topics and the emergent semantic narrative arc.[80] Parrish is interested in creating artifacts that do afford this kind of reading and that give rise to narrative as an epiphenomenal or emergent property as opposed to something that has been explicitly encoded.

The act of generation is also something that blends into the editing process and the formatting of the textual output for print. Parrish ran her random walk algorithm through the phonetic vector space around 20 times, and selected a result that looked like it had a somewhat interesting beginning and ending without too much repetition. Due to the stochastic nature of the process, a wide variety of traversals would produce very different aesthetic feelings. This is a common mode of curation and editing utilized by author/programmers who write something that will end up as a physical book.

Pagination

The concept of not editing the output is something that is important to Parrish artistically. She says that, for her, "what's interesting about working with computation is that the decisions that you're making are in the program."[81] This speaks to a particular poetics of programming and

[79] H. Porter Abbot, *The Cambridge Introduction to Narrative* (Cambridge University Press, 2008), 19.
[80] Parrish, Appendix A.
[81] Parrish, Appendix A.

computation, but also a subset of creative techniques within the realm of machine learning specifically. Similar to Sol Lewitt's instructional wall drawings, Parrish is interested in the practice of making poetic decisions that are essentially a list of instructions. Once you start to edit the computational output, the work moves toward more conventional poetic decisions that aren't as interesting to Parrish, decisions that she "[doesn't] have a lot of confidence in making." She says that "the strength of that opinion rests a lot on [her] lack of self-confidence about [her] ability to write a conventional poem."[82] She feels like she is very good at making the algorithmic decisions, and tends to stick to those, speculating that if she had a background in conventional poetry she might be more interested in explorations of co-creation.

Despite being very dedicated to not editing after the act of generation, other than choosing among generated outputs, some editing of the text did occur. When reading through the proof of the book, Parrish came across an antisemitic slur that slipped through the word filters, provided by a poem that Coleridge wrote to a friend. "You just can't trust these 19th century Englishman to not have terrible attitudes about things," Parrish said as she recalled the cumbersome process of contacting the publisher, dumping the first run of the book, and paying for the second printing. She did not rerun the algorithm because she liked the output she had chosen and sent in a new manuscript with the one line removed. Referring to the final book, Parrish said, "I'm OK being responsible for all of the language in this text, and that's what's important to me."[83]

In a review of *Articulations* for *Rhizome*, Brendan C. Byrne writes that the book, "throws

[82] Ibid.

[83] Ibid.

the reader, again and again, onto the essential dilemma of the nature of authorship."[84] But this editorial act points out the unique questions that can arise not from thinking about the computer as an author, but thinking about the nature of computer-generated literature in relation to the human author of a system and its resulting output that necessarily needs to analyze or manipulate some kind of source text. "On the cover it doesn't say Allison Parrish and PoetryBot3000, it's just me, it's my work, I wrote it, I wrote the programs to write it," she said.[85] Not all author/programmers take this point of view, and some do attribute authorial credit to the machine, as was the case with the 1984 *The Policeman's Beard is Half Constructed*. This taking responsibility for the output and not outsourcing authorial credit or blame to the machine is an increasingly important question as cultural conversations around the perceived objectivity of algorithms and computation are more and more focused on revealing and critiquing the social and human factors that affect and are affected by computer programs as they act in the world.

In this system, written by Parrish, we can trace the material trajectory of text from the writers of the original publications of the printed books that were then typed up and distributed online by Project Gutenberg volunteers. These plaintext files were then scraped into a dataset of Project Gutenberg entries, which was then mined for lines of poetry that were arranged in a vector space based on phonetic similarity. Parrish sent a random walk algorithm strolling through the vector space multiple times. The outputs of these walks were compared, and one was chosen to be published, with one line of the chosen output eventually deleted by the author.

As one example of the millions of lines of poetry that were part of this assemblage, we can examine two unique lines in a template for the Sloka, "a distich of two sixteen syllable-lines,

[84] Brendan C. Byrne, "An Excess of Consciousness," Rhizome, July 10, 2018. https://rhizome.org/editorial/2018/jul/10/an-excess-of-consciousness-1/.
[85] Parrish, Appendix A.

divided at the eighth syllable,” that shows up in the Project Gutenberg etext Preface to *Nala and Damayanti and Other Poems*, by Henry Hart Milman.[86]

```
u u u u | u - - - | u u u u | u - u -
- - - - |      u | - - - - |      u
u u u u | u - - - | u u u u | u - u u
- - - - | -    u | - - - - |      -
```

This placeholder template was recognized as lines of poetry in the database system that Parrish developed. Then, because the second and fourth lines consist of more than 25% punctuation characters, they are not included in A Gutenberg Poetry Corpus, adding only the first and third lines to the dataset.

```
{'s': 'u u u u | u - - - | u u u u | u - u -', 'gid': '19529'}
{'s': 'u u u u | u - - - | u u u u | u - u u', 'gid': '19529'}
```

These lines stick out compared to the majority of the lines in *Articulations*, but after gaining an understanding of the computational processes behind the work, the textual contexts of the line, and the phonetic logic of the system, we can read the machine as it reads this line in the context of its poetic gradient.

> “You warm your hands Where are you — you — you — you. U u u u | u - - - | u u u u | u - u - u u u u | u - - - | u u u u | u - u u I saw you in the street.”

By calling attention to the mechanisms of machine learning logics, *Articulations* both resists the obfuscating seamlessness that dictate the design of daily interactions with AI systems, while creating a seamlessness of its own. This moment of phonetic disintegration across

[86] Henry Hart Milman, “Nala and Damayanti and Other Poems,” edited by Sankar Viswanatha,. Project Gutenberg, October 12, 2006. http://www.gutenberg.org/files/19529/19529-0.txt.

non-word characters is one of the moments that best illustrates the mechanisms behind *Articulations*. Through a close look at the writing process of *Articulations*, we begin to understand a new kind of poetic relationship that involves specific writerly practices regarding datafication, vectorization, generation, and pagination.

1 THE ROAD

1 the Road by Ross Goodwin is an experiment in mechanical tone, spontaneous and automatic writing, the creative subversion of institutional and capitalist technologies, and an attempt to write a road trip novel with a car. Despite these subversive goals, the project itself was part of a collaboration with Google, one of the main developers of institutional and capitalist technologies. This is representative of the unique positioning and creative negotiations that practitioners of AI-generated art are involved in. Goodwin has been employed by Google, where he developed his creative practice and helped "break the mold on how engineers think and use creativity to make new discoveries in a less-creative field."[87] The book was supported by Google Arts & Culture and published by Jean Boîte Éditions (JBE) Books, a French art book publisher with collections that emphasize artistic practices that are unique to the digital age. *1 the Road* is included in JBE Books' Art + Machines Collection, which consists, currently, of just this book.[88] This positions *1 the Road* as the first book in this collection and suggests both an anticipation of and commitment to literary work that explores the intersection of machines and creative expression.

Goodwin's gonzo experiment began in New York, with a black 2017 Cadillac XTS fitted with an Axis M3007 surveillance camera on its trunk and a GPS unit on its roof. Inside the car, a

[87] Goodwin, Appendix B.
[88] "Art + Machines," JBE Books. https://www.jbe-books.com/collections/art-machines.

Razer Blade laptop powered an internal microphone that captured audio spoken in the car and a built-in clock that provided the structure of the time-stamped road trip travel log. The data gathered by these devices was intermittently seeded into an LSTM recurrent neural network that generated a short poetic statement at a particular moment in space and time. On the four-day journey from New York to New Orleans, this system captured data roughly every 30 seconds and printed its entry onto a scroll of wide-format thermal paper from a portable Datamax O'neil printer mounted in the car. The drive resulted in 11 full scrolls, each consisting of 127 feet of thermal paper, echoing Jack Kerouac's legendary 120-foot first draft scroll of *On the Road* that was pieced together with cut-up drawing paper. Compared to *Articulations*, the selection of hardware at every level of Ross Goodwin's surveillance car interface is emphasized as part of the poetic and writing process of *1 the Road*. The work of *Articulations*, while dependent on platforms such as Python and certain hardware to perform necessary computation, is more focused on the development of the model, regardless of coding language or hardware interface.

This emphasis on interface is consistent across many projects in Goodwin's practice, which centers around what he calls "narrated reality," or the development of "new forms and interfaces for writing, enabled by machine intelligence."[89] One of his earlier NR (to keep with the naming convention of VR, XR, and AR) experiments was a project called word.camera. This project is an NR device that takes pictures and automatically captions the resulting images with poetic language that is then printed on thermal paper alongside an ASCII version of the image. The entire captured scene is rendered into text. This project underwent many iterations and used different types of neural networks and machine learning frameworks in its different

[89] Ross Goodiwn, "Narrated Reality," 2016. https://narratedreality.com/.

instantiations. This word.camera project was also part of a series of NR interfaces that Goodwin created in 2016, which included not only a camera that narrates images, but also a compass that narrates location, and a clock that narrates time.[90] Since these projects and the publication of *1 the Road*, Goodwin has further developed this technique of making objects narrate. In a collaboration with stage designer Es Devlin and Google Arts & Culture called "Please Feed the Lions," Goodwin developed an interface to give voice to the lions in London's Trafalgar Square. [91] The system accepted donated words from passersby and online participants that were seeded into an LSTM that was trained on 19th century poetry. The generated lines were then displayed in the mouth of the lion and projected onto the lion's body. The final collective output of the poem was also published online.[92] *1 the Road* is an interesting stop in Goodwin's practice of Narrated Reality and the only project of his that has been printed in the form of a codex book. The system that generated *1 the Road*, which Goodwin calls "wordcar," includes code from his past projects and is the step in his practice that followed the creation of the separate NR camera, compass, and clock interfaces.

[90] Ibid.

[91] "The Technology and People behind Please Feed the Lions," Google Arts & Culture, 2018. https://artsandculture.google.com/exhibit/the-technology-and-people-behind-please-feed-the-lions/uQJy-TQgHNPkJw.

[92] "The Lion's Roar: 23rd September," Google Arts & Culture, 2018. https://artsandculture.google.com/theme/the-lion-s-roar-23rd-september/KQIytu3ZhNlDKw.

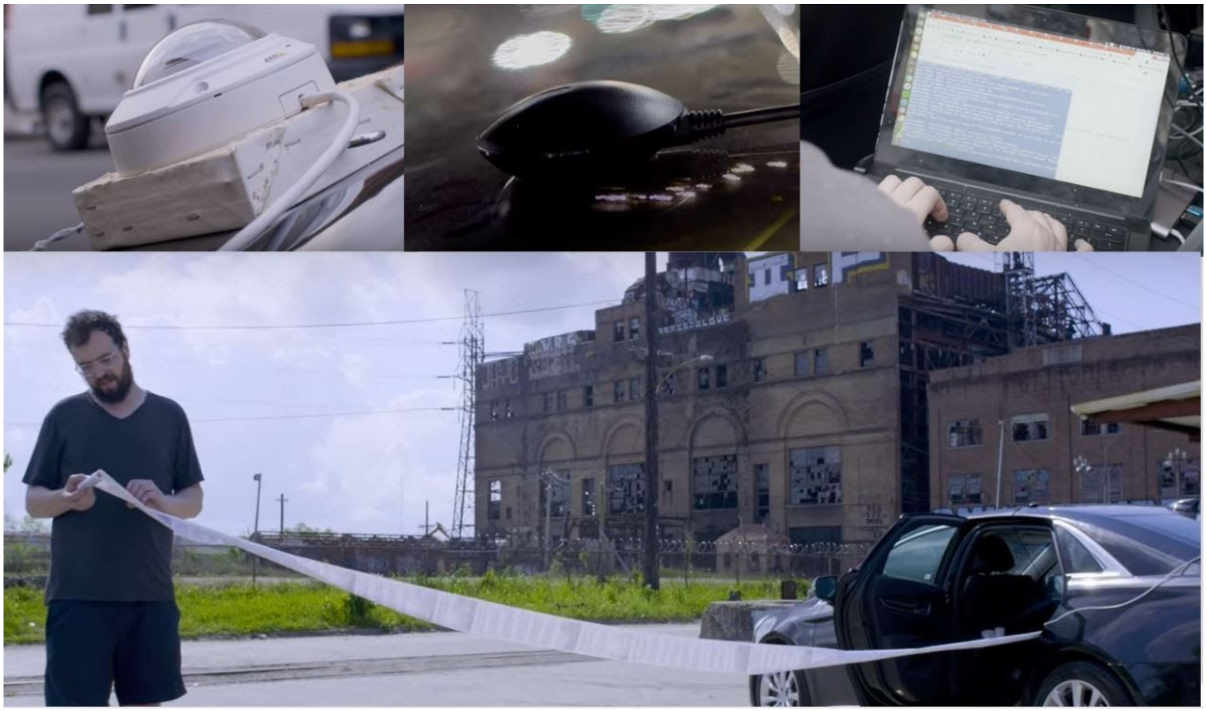


Figure 9. The surveillance car apparatus used to generate 1 the Road. (Top, From left to right): the surveillance camera, the GPS unit, the laptop inside the car with the internal clock and microphone. (Bottom): Ross Goodwin holding the printed scroll of textual output coming from inside the car.

Datafication

Goodwin has his own archive of strange and specific data sets that he has been curating for years. He remains in conversation with this data, and knows what the datasets are going to say, or what they usually sound like. *1 the Road* is mainly generated using an LSTM that is trained on a dataset that he curated for his friend, the Polish painter Roman Lipski, who is also extensively involved with AI experiments in creativity. The corpus contains all of Lispki's favorite books, which are mostly bleak Eastern European novels that matched the tone that Goodwin was looking for. *1 the Road* was inspired not only by Jack Kerouac, but by countercultural American road narratives in general, including the work of Tom Wolfe and Hunter S. Thompson. After reflecting on the project, Goodwin said that he was glad that none of these texts were included in the actual training data because he sees that as a kind of cheating.[93] This is a common reaction to computer-generated text that Goodwin sees as a viable criticism.

With practices and accusations of fraud in mind, Goodwin emphasizes that, "You have to be transparent with systems like this, because otherwise you are just sort of a glorified magician." Sometimes Goodwin feels like a magician anyway due to the sometimes intense or perplexed reaction that he gets in response to his work. Goodwin says he, "always tr[ies] to be very up front about what's happening." He continues: "I guess, sometimes I feel like I'm peddling something, like the tonic dealers of old, like a hundred years ago, who had their medicinal tonics that cured everything, and sometimes when I get near product people who want me to do things that are more commercial, and when I consider doing things like that, I start to feel more like a tonic peddler because in a way, it's not that what I'm selling is fake or not legitimate, it's more like any marketing around this, because of the way that AI is perceived in the present as sort of like a panacea, is going to make it seem like, oh it's a miracle, look at this text come out! Oh my God! It will solve all your problems! It'll solve your writer's block! It's very sensational, it lends itself to sensationalism, just the idea that something that isn't human is capable of writing even one coherent sentence."[94]

Transparency and originality are key concerns for Goodwin and one of the reasons he was initially drawn to machine learning rather than other, less computationally intensive techniques for text-generation.[95] Goodwin sees the ability to create original texts rather than parrot back chunks of a corpus as one of the main advantages of LSTMs. When he finds

[93] Goodwin, Appendix B.

[94] Ibid.

[95] Ross Goodwin, "Adventures in Narrated Reality," Medium, Artists Machine Intelligence, June 9, 2016. https://medium.com/artists-and-machine-intelligence/adventures-in-narrated-reality-6516ff395ba3.

something in the output of one of his systems that he is particularly struck by, he will search the phrase or line in Google to make sure that it is in fact original. While still fundamentally a combinatorial text-generator, the LSTM, especially when trained at the character level rather than the word level, is one example of the genre of combinatorial writing no longer holding much meaning. Once the unit of the cut-up is no longer large sections of pages as Burroughs suggested or chunks of text like lines of poetry, but rather any individual character in a very extensive corpus, the defining feature of such a system is no longer the way in which characters, words, or phrases are combined, but the numerical and vectorial relationships between each datapoint in the vector space of the RNN.

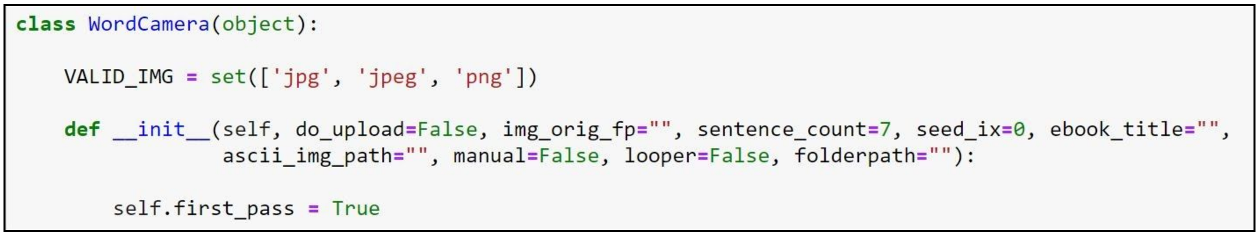
Vectorization

The process of vectorization in *1 the Road* is also not a main emphasis of Goodwin's machine learning project as it is with *Articulations*, but it is still a necessary and definitive step in the writing process. The use of an LSTM or any neural network for the purposes of text generation necessitates that some type of linguistic data are represented as vectors. Specifically in the case of torch-rnn, linguistic data is represented at the character level and trained not on sequences of phrases or words, but letters, punctuation, and any other types of individual characters. Each character is represented initially as a co-occurrence vector that is then fed into an LSTM, powered by torch-rnn, which was a groundbreaking system when it was made publicly available by Stanford computer science student Justin Johnson in 2016.[96] The process of training incrementally changes the values in each vector, adjusting the weights and biases, such that the model will be able to accurately predict sequences of characters in a given test corpus.

[96] Justin Johnson, "Jcjohnson/Torch-Rnn," GitHub, February 16, 2016. https://github.com/jcjohnson/torch-rnn/blob/master/doc/modules.md.

Once the model is trained, each vector in the resulting vector space consists of numbers that are relational to the statistical likelihood that the character will follow another character.

1 the Road uses two types of RNNs in its text generation system. One is the LSTM, which interacts with the audio, GPS, and time data, and eventual image caption, while the other is a convolutional neural network, or CNN, which is typically used for analyzing and classifying image data. The CNN used in wordcar is part of densecap, which was also released by Johnson and his lab at Stanford. The densecap program is a publicly available model that uses a CNN to automatically caption images in natural language.[97] This classification, in the form of textual output, is then sent to an LSTM, which in turn generates a poetic caption, much like word.camera. The main chunk of code that brings all of the aspects of the system together in wordcar is actually a repurposed version of the code from the preceding iteration of word.camera.



```
class WordCamera(object):

    VALID_IMG = set(['jpg', 'jpeg', 'png'])

    def __init__(self, do_upload=False, img_orig_fp="", sentence_count=7, seed_ix=0, ebook_title="",
                 ascii_img_path="", manual=False, looper=False, folderpath=""):

        self.first_pass = True
```

Figure 10. *The WordCamera object in the main_rb_car.py file in the wordcar system and its initialization parameters. Despite being called "WordCamera," this is where all of the different types of data are processed and seeded.*

Generation

The bulk of Goodwin's writerly work for *1 the Road* occurs in the generation stage, or during the creation of textual output, which involves an elaborate system for seeding the neural

[97] Justin Johnson, "Jcjohnson/Densecap," GitHub, June 13, 2017. https://github.com/jcjohnson/densecap.

network models in a way that interacts with all of the sources of data captured by the wordcar interface. While it is true that *1 the Road* incorporates data from sources other than the training corpora used in the LSTM and CNN models, these data are part of the generation procedure rather than a preprocessing step, although they are each specifically parsed by functions that were written as part of the wordcar system. Goodwin's wordcar interface is outfitted with four modes of collecting data: the surveillance camera, the GPS unit, the laptop microphone, and the laptop's internal clock. These sources of visual, locative, audio, and time data prioritize certain modes of experience that are situated in an understanding of being that is constantly tracked, defined by, and in interaction with digital archives. Søren Pold and Malthe Stavning Erslev call this "data-realism," and argue that third-wave electronic literature is defined by the representation and reconstruction of metainterfaces, or spaces of "data collection, standardization, commodification and redistribution."[98] Each source of data is both highly personal in its proximity to and surveillance of Goodwin's road trip and extremely impersonal in its standardized corporate manifestation. The data are everywhere, but the text and generation process of *1 the Road* highlights Goodwin's journey through it all, similar to artists who travel to create GPS drawings.[99]

The GPS unit on the car tracks its latitude, longitude, altitude, and speed. The altitude and speed data are originally recorded in metric units, which Goodwin explicitly converts into imperial measurements. The Foursquare API, which hosts a dataset of locations that consists of

[98] Søren Pold and Malthe Stavning Erslev, "Data-Realism: Reading and Writing Datafied Text," electronic book review, March 5, 2020. https://electronicbookreview.com/essay/data-realism-reading-and-writing-datafied-text/.

[99] Chris O'Brien, "Meet the Japanese Artist Who Made the World's Largest GPS Drawing for His Marriage Proposal," VentureBeat, November 20, 2014. https://venturebeat.com/2014/11/17/meet-the-japanese-artist-who-made-the-worlds-largest-gps-drawing-for-his-marriage-proposal/.

the name, category, latitude, and longitude of venues, is used to collect the name and category of the nearest location. This data is then parsed into a string that has the structure: [Venue Name]: a [Location Category] in [City Name]. During the generation process, if a valid venue cannot be located, this seed string instead takes the format: [Latitude] N, [Longitude] W, at [Altitude] above sea level, at [Speed] miles per hour. The time data, taken from the laptop's internal clock, is parsed in a similar fashion, but with more variation. A separate function provides a template for generating a valid string such as, "It was nine seventeen in the morning,"[100] using a template along the lines of "[Sentence beginning option] [Hour] [Minute] [Time of day descriptor]" with multiple options for each phrase fragment and exceptions for special cases like noon or midnight.

The audio recording process, when run, records sixteen seconds of audio from the Razer Blade laptop's internal microphone and transcribes it using a program available on GitHub called Audiogrep, which in turn is built upon the Carnegie Mellon University Speech Recognition Toolkit.[101] The transcription is split by lines, and only the first line is kept, capitalized, and used as a seed for the LSTM. Finally, the image data is captured from the surveillance camera positioned on the outside of the car. An ASCII version of the captured image is created and printed while the image itself is sent to densecap, the other neural network model that is used in the wordcar system, to generate a caption that can be used as a seed.

This is not the order in which this process of seeding occurs. Rather, the choice of seed is made roughly every thirty seconds, and is randomly selected from the four sources of data. However, the probability distribution is not entirely equal. First, the system has a 33% chance of selecting the image data as the seed choice. If the image data is not selected, then the audio, time,

[100] Goodwin, *1 the Road*, 41.

[101] Nickolay Shmyrev, "CMUSphinx Open Source Speech Recognition," CMUSphinx Open Source Speech Recognition, Carnegie Mellon University. https://cmusphinx.github.io/.

and GPS data each have a 33% chance of being selected. This means that in total, the audio, time, and GPS data each have a 22% chance of selection, while the image data is slightly more likely, at 33%. This, along with the parsing of the seed data, is the part of the system that is combinatorial and more along the lines of past traditions of AI and computer-generated literature. While still exploring the possibilities afforded by vector representations of linguistic data and machine learning, *1 the Road* is much more similar to the rule-based *The Policeman's Beard is Half-Constructed* than *Articulations*. *Policeman's Beard* also featured images alongside the text in the form of collages created by Joan Hall, an artist and friend of RACTER's programmer Bill Chamberlain. *Policeman's Beard* was a rule-based project and, in a way, the rule-based seed parsing in *1 the Road* is allusive to this literary and computing history. While reading the code alone is not a substitute for the text and should not make up the entire reading experience of AI-generated literature, this type of relationship is one that can only be analyzed through looking at the code.[102]

Pagination

"As a practitioner of this sort of work," Ross says, "it's very hard to find a collaborator, whether it's a publisher or a filmmaker, who is not going to say things like 'Oh this paragraph doesn't work, we can't do this, delete.' There are always compromises." While some editing of the text by the publisher did happen, it was purely subtractive, and by design. Goodwin admits, "I didn't want to give any publisher or anybody who got the manuscript the opportunity to heavily curate down to a really really high-quality book [...] It was about pressing a button and making it happen rather than building in all of this extra work that would happen later."[103]

[102] John Cayley, "The Code Is Not the Text (Unless It Is the Text)," electronic book review, October 9, 2002. https://electronicbookreview.com/essay/the-code-is-not-the-text-unless-it-is-the-text/.
[103] Goodwin, Appendix B.

Through the editorial process, generated timestamps were subtracted, and other decisions were made such as changing the typeface of the text to Junicode, a proportional, serif font that perhaps was chosen to allude to the "literary" aspects of the text in contrast to the font that was used during the printing process in the car, which was a monospace font that is typical of programming environments.

Additionally, records of human labor are made visible through errors like the typo on page 83. The timestamp at 13:55:10 on March 26, 2017 reads "The time was five minwutes until two o'clock in the afternoon. The conversation was still still at the back of the street, and the sun was still frozen in the sun."[104] This passage stands out because the time data was parsed in a rule-based way that would not allow for any misspellings of the word "minute." Even if the typo were in the section of the text that was generated by the LSTM, the model does not make these types of spelling mistakes either. The nonsensical aspects of the text emerge from grammar inconsistencies or word orders that are not part of conventional English, never from the lack of ability to spell a word correctly. The human error that is apparent in the snippet of otherwise mechanically "correct" language introduces a uniquely human mistake that positions the human perspective in contrast to that of the machine. Within the text itself, it also adds the image of human movement into a scene that is otherwise "still" and "frozen."

The first sentence of Goodwin's author bio plainly states that he is, "not a poet," but a, "writer of writers."[105] This negation of the title of poet is a rhetorical move similar to that of René Magritte's *The Treachery of Images*, in which a painting of a pipe is labelled as "not a pipe," or David Markson's novel *This is Not a Novel,* which is an assemblage of seemingly

[104] Goodwin, *1 the Road*, 83.
[105] Ibid., Back cover.

isolated and self-contained anecdotes, datapoints, and statements.[106] This specific type of rejection of traditional classification points to the influence of surrealism on author/programmers more generally and is in line with Goodwin's, and Parrish's, goal of "break[ing] literature in interesting ways."[107] However, Parrish definitely considers herself a poet and does not try to break away from traditional authorial categories in this way. *1 the Road* is very similar to Markson's novel in its structure of self-contained absurdities, but it is perhaps even more of a traditional novel in that it is clearly labeled as such[108] and portrays a sequential representation of events through the mediated road trip over the course of four days. The *story* of the road trip lies in the sequential nature of the time, location, image, and audio data while the *narrative discourse* is represented through surreal scenes that sometimes conjure up a poetic image and create a small story in themselves.

"The time was six thirty-one in the evening and Teddy was already in the kitchen"[109] is an example of a scene that makes sense. As the car drives by, equipped with the omniscience of data, it can see into imaginary lives. Somewhere, an hour's drive out of Crawfordville, someone called Teddy is making dinner early. Sometimes, the scene is surreal, and conjures up interesting relationships and provokes questions: "A cat-driver, a dog walking down the highway."[110] Why does the cat get to drive and the dog have to walk? The addition of the ellipses as an option for ending a sentence also creates the impression that the car passed by too quickly to fully get a handle of what was going on: "Mall at Stonecrest: a shopping mall in Lithonian barn, and the

[106] Young Hee Kim, "David Markson's This Is Not a Novel: A Deviation from Traditional Fiction Writing," *Studies in English Language & Literature* 43, no. 2 (2017): 27–47. https://doi.org/10.21559/aellk.2017.43.2.002.
[107] Goodwin, Appendix B.
[108] Jean Boîte Éditions, *A gonzo AI takes the road,* Press release.
[109] Goodwin, *1 the Road*, 101.
[110] Goodwin, *1 the Road*, 102.

sound of cars, all the faces were clasped by a strange stranger in an attempt to find a way of getting a bit of …"[111] but the car is already past the mall, and we don't get to know what the people listening to the stranger were trying to get. The closing time stamp of the third day also brings the reader into the road trip journey. After many hours of driving, an average of almost twelve hours a day over the course of three days, the car pulls into a parking garage in New Orleans: "LSUHSC Parking Garage: a parking in New Orleans, a tall gallon garage with a stucco striped strip of water had been shredded and the dark color of a corn stain."[112] Each day of text generation is separated in the book by a blank white page, which the reader begins to associate with rest and sleep between days of travel. After pulling into the garage at the end of Day 3, the blank white page comes as a relief and a moment of rejuvenation, from the length of the journey, from being stuck in a car for so long, from the difficulty of making sense of a challenging text. This is an emotional part of any road trip journey and serves the gonzo narrative well. The hardest part is over and you get to enjoy the last day of driving and say goodbye to your journey. Day 4 is only three hours (and 6 pages) long.

RERITES

David Jhave Johston is a digital poet, artist, and scholar working in computational media. His project *ReRites* is the product of one year of sustained practice involving an AI writing experiment. Johnston woke up at 6am for six to seven days a week for an entire year and edited the output of various text-generation systems powered by different types of recurrent neural networks. He edited this output for two hours each day, and collected every month's worth of poems in an individual printed volume. Each month, the model changed, either in its training

[111] Goodwin, *1 the Road*, 103.
[112] Goodwin, *1 the Road*, 132.

corpora or the model itself. Sometimes Johnston used PyTorch, an open source deep learning framework created by Facebook, sometimes Tensorflow, a similar framework created by Google, and sometimes he used Salesforce's open source LSTM model. Johnston's year of work resulted in twelve books of poetry that were printed and sold as a limited edition box set from Anteism Books in 2019. Books 1-12 were also accompanied by a "Book 0" that includes eight contextualizing essays from poets and scholars, as well as an introduction and an author note from Johnston himself. Twelve of these box sets exist in the world, and they are each priced at USD $1000. The audience for this particular collection is made up of mostly rare book libraries and other collectors and is not available for much of the wider public to read. The book that I am focusing on is *ReRites - Raw Output / Responses*, the more commercial version of the project that is priced at USD $18 and is meant to get into people's hands and provide a sample of the poetry and the project in an affordable format.

ReRites - Raw Output / Responses includes the essays that make up Book 0 in the *ReRites* box set, along with 60 poems selected from the more than 4500 pages of poetry within the twelve volumes (between four and six poems were chosen from each book), and 23 pages of examples of "raw output," which is described as having, "no human intervention."[113] This smaller paperback book is emblematic of the larger work as a whole and provides extensive contextualizing information for the poems themselves as a way to demonstrate the amount of human labor that went into the writing process. The raw output, without human intervention, is intentionally placed alongside the poems that resulted from Johnston's editorial process, which he called, "carving." The giant blocks of AI-generated text were treated as a medium to cut away

[113] David Jhave Johnston, Allison Parrish, Johanna Drucker, Kyle Booten, John Cayley, Lai-Tze Fan, Nick Montfort, Mairéad Byrne, and Chris Funkhouser, *ReRites - Raw Output / Responses*. Responses introduced and edited by Stephanie Strickland, (Quebec: Anteism Books, 2019), 69.

and shape. Just looking at the twelve volumes in the *ReRites* box set provides a visualization of the ability of neural networks to augment human creativity and the relentless endurance of the machine.

Despite the specification that “no human intervention” was involved in the generation of the text, the sample of Johnston’s code that accompanies the raw output in the book explicitly demonstrates authorial decisions that went into the project. At the beginning of running each model and generation system, a series of strings is printed. These strings specify the type of neural network chosen, the dimensionality of the embedded vectors, the number of training epochs, and a number of other metrics that were experimented with and fine-tuned to produce different sorts of output. As the program runs, Johnston provides commentary and asks himself to wait: “Initializing. Please be patient.”[114] The strings also state the date of generation, and that the system will infinitely generate poems of 88 words each until stopped. Details about the training corpus are also provided, including the number of lines in the text file and the sources that those lines were scraped from. There is also necessary human intervention in the decision to eventually stop the program and the generation.

[114] Johnston, *ReRites - Raw Output / Responses*, 70.

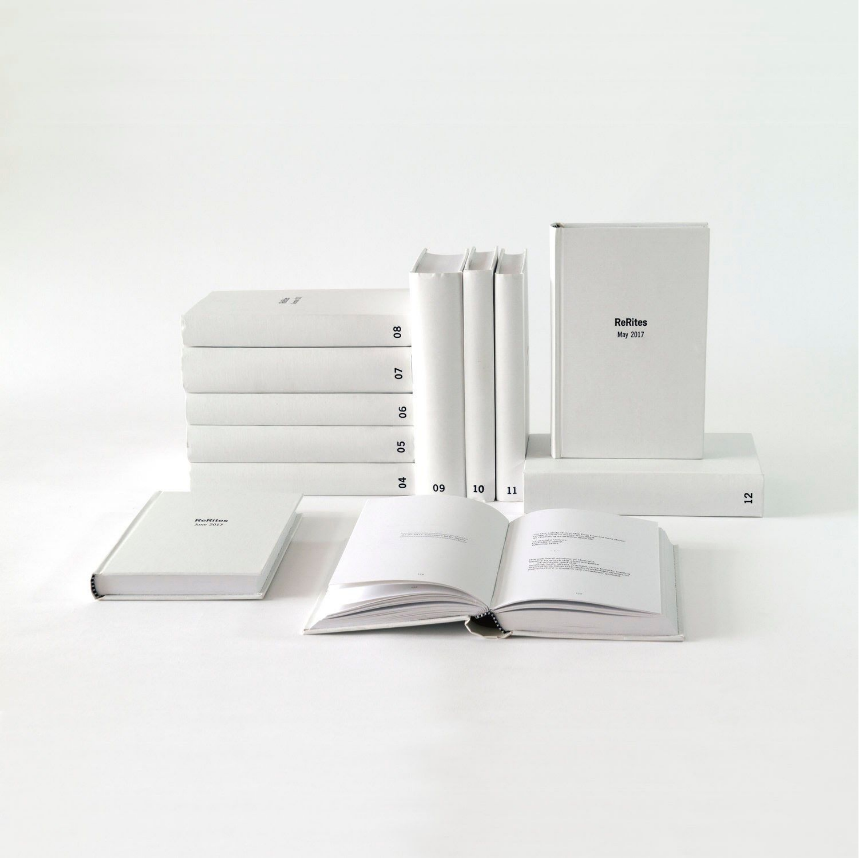


Figure 11. *The limited edition box set of ReRites. With the assistance of AI, David Jhave Johnston wrote 12 volumes of poetry in one year.*[115]

In the context of Johnston's multimedia work, this text-based print project stands out. He began his career as an artist, writer, and painter, before returning to school in computer science and then digital media. He began experimenting with the ability to take an artistic practice and explore it in a software ecosystem, creating works of poetry mostly in Flash and for the web. Much of Johnston's practice involves experimenting with animated textuality that strays away from the lineated page and focuses on more dynamic, interactive, and multimedia ecosystems. In his 2016 book *Aesthetic Animism*, Johnston writes that upon the advent of digital poetry, "the

[115] Anteism Books, 2019. https://www.anteism.com/shop/rerites-david-jhave-johnston.

page becomes vestigial."[116] Despite the lack of emphasis on printed work in his practice, Johnston said that "the neural networks are so strong at creating language that [ReRites] very swiftly became a pure writing project," and that this focus on pure text lent itself to the medium of the printed book.[117] As a theoretician of digital media, Johnston was and is well-acquainted with the history of computer-generated literature. What set machine learning and neural networks apart from previous traditions of creative text-generation and what drew him to the technologies was the ability to encode linguistic relationships without explicitly hard-coding rules of language in a laborious way. He notes that rule-based systems don't allow for much flexibility or edge-cases of language, and, for Johnston, "the edge cases are kind of where poetry happens."[118]

Datafication

Much in the same way that Goodwin is constantly creating and maintaining datasets to train certain models, during the duration of the project, Johnson was always "foraging" for source texts to compile as training data for his system. For *ReRites*, he was going after something that replicated 20th century avant-garde conceptual poetry and foraged for language that might achieve that aesthetic. It is possible to train a neural network on the works of one prolific author or on one specific type of literature in order to mimic it, but Johnston, similar to Goodwin, was trying to use less conventional corpora to build something that would generate text that somehow alluded to a certain tone or experience. For this project, he was also looking for language that would be interesting enough to sustain his interest for an entire year. Instead of training his

[116] David Jhave Johnston, *Aesthetic Animism: Digital Poetry's Ontological Implications* (Cambridge, MA: MIT Press, 2016), 3.
[117] David Jhave Johnston (author of *ReRites*) in discussion with the author, March 24, 2020. See Appendix C.
[118] Johnston, Appendix C.

models purely on, for example, iambic pentameter or Shakespearean prose, he included a range of source texts that featured modern poetry, but also song lyrics, news articles, research articles, and quotes that he had enjoyed and collected over the years.

Each month was a collaboration with a machine that had a slightly different character, and that augmented his writing in different ways. As the months went on, Johnston also began incorporating previous months of the poetry in the *ReRites* project. Despite not depending on networked technologies for the process of generation, this is a prime example of Lori Emerson's readingwriting. The system "reads" a large corpus of human-written texts and generates some output. This generation is an instance of the machine then writing Johnston's reading. After carving the output and creating more poetry, this writing is then fed back into the system, which in turn reads Johnston's writing in a feedback loop of computationally-mediated understandings and representations of text. While the system doesn't necessarily need access to a networked computer, this process is emblematic of the computational and algorithmic divisions of labor in a networked society. As Kyle Booten notes in his response, "Harvesting ReRites," Johnston is both "above and below the API,"[119] embodying both the managerial and working class of the information economy. To be below the API, or "Application Programming Interface," is to be told by computers what to do on a daily basis.[120] Being managed by an app if you are an Uber driver is an example of this. To be above the API means that you are in the position of designing the computational infrastructure of society, developing the Uber app, for example, and in turn

[119] Johnston, *ReRites - Raw Output / Responses*, 111.

[120] Anthony Wing Kosner, "Google Cabs And Uber Bots Will Challenge Jobs 'Below The API'," Forbes, February 4, 2015. https://www.forbes.com/sites/anthonykosner/2015/02/04/google-cabs-and-uber-bots-will-challenge-jobs-below-the-api/#403bfce569cc.

directly impacting and controlling the lives of many. Johnston turns this process in on himself, both managing and obeying, in a recursive and voyeuristic system.

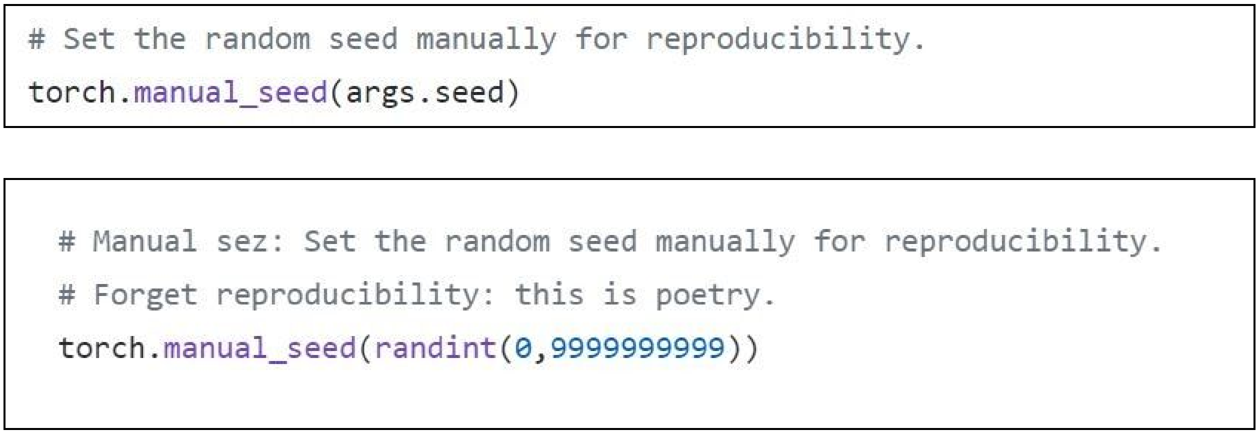
Vectorization

Over the course of the year, Johnston experimented with three different open source language models: a TensorFlow implementation of Google DeepDream's WaveNet program, (one of the first models that he used), a PyTorch implementation of a word-level LSTM (which makes up the bulk of the project), and during the final few months, he forked his PyTorch code to include a new model from Salesforce Research's updated version of the PyTorch word-level LSTM.[121] These models vectorize linguistic data at the word level, rather than the character level like torch-rnn, used in *1 the Road*. However, both models are developed using the same open source machine learning library, Torch, which is developed mostly in the Lua and C programming languages.[122] The main LSTM model used in *1 the Road* was part of torch-rnn, a neural network library released by a student at Stanford in mid-2016. A few months after that, PyTorch was released. PyTorch is an open source Python implementation of the Torch machine learning framework from Facebook's AI research lab. Each of these models has their own embedding process built in in order to quickly format data to feed into a neural network. As opposed to Parrish's very laborious and novel system for phonetic vectorization, these corporate tools provide a relatively seamless interface for the embedding and vectorization steps necessary to work with machine learning.

Johnston's experiments with AI began in a very corporate environment in 2014, when he took a course in data science through General Assembly, a bootcamp for teaching business

[121] David Jhave Johnston, "Jhave/RERITES-AvgWeightDescentLSTM-PoetryGeneration," GitHub, June 30, 2018. https://github.com/jhave/RERITES-AvgWeightDescentLSTM-PoetryGeneration.
[122] "Scientific Computing for LuaJIT," Torch. http://torch.ch/.

professionals “practical”[123] technology skills. Johnston recalls “sitting there with a bunch of [...] demographic technicians from bubblegum corporations or financial traders who were all trying to optimize the monetary flow of their various networks,” being interested in messing with language and generating novel poetic text rather than improving profits or learning practical business technology skills.[124]



```
# Set the random seed manually for reproducibility.
torch.manual_seed(args.seed)
```

```
# Manual sez: Set the random seed manually for reproducibility.
# Forget reproducibility: this is poetry.
torch.manual_seed(randint(0,9999999999))
```

Figure 12. *(Top) The manual seed code in the PyTorch word language model documentation, (Bottom) The seed in Johnston’s fork of the PyTorch word language model*

Although the code that Johnston used and changed for *ReRites* is open source, each aspect of the project is still deeply affected by corporate logic. Each of the models used is associated with a different megacorporation. The computationally intensive training of Johnston’s models was assisted by the NVIDIA Academic Grant Program, which provides graphics processing units (GPUs) to full-time faculty and researchers and granted Johnston an NVIDIA TitanX GPU. Working with AI, acquiring the necessary hardware, and experimenting with the latest state-of-the-art approaches is difficult, if not impossible, to separate from

[123] Ilya Pozin, “20 Entrepreneurs Shaking Up New York's Tech Scene.” Forbes, September 15, 2014. https://www.forbes.com/sites/ilyapozin/2014/09/13/20-entrepreneurs-shaking-up-new-yorks-tech-scene/#1078c57029e7.
[124] Johnston, Appendix C.

corporate tech giants, notions of labor, privilege, and even the organization of society as mediated by software.

Generation

Using Johnston's system, just two hours and forty-eight minutes of waiting can generate 12,000 lines of novel poetry.[125] Whereas the generation system behind *1 the Road* features an elaborate process of creating seed strings of text and using them to generate phrases in the vein of predictive text memes[126] and performance pieces,[127] the text generation system for *ReRites* is one that is concerned with creating massive amounts of output. Ultimately, the generation stage is focused on the manufacturing of textual material to then be manually processed by Johnston. This approach is in opposition to that of Parrish and Goodwin. Goodwin specifically did not want the textual output of his system to be edited down into something that was conventionally "high quality," whereas for Johnston, the editing down and end quality of the poems was of primary concern. At the beginning of each month, enough text had to be generated for at least 48 hours of sustained carving.

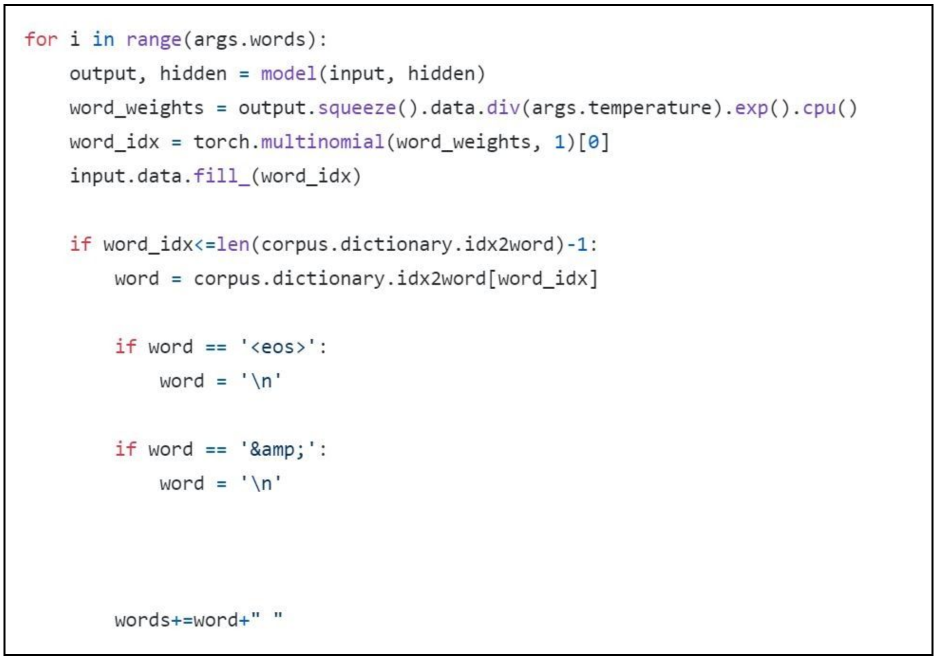
Even though the generation process was largely one of waiting, Johnston still made certain decisions about the format of the output. The training and seeding of the neural network is not the only computational work that is done. Many of Johnston's generation programs run an infinite loop until exited. Johnston sometimes chose to generate individual poems that were 88 words long. This programming decision is similar to that of Parrish who had to specify what her

[125] David Jhave Johnston, "Saved Output." GitHub, April 19, 2017. https://github.com/jhave/pytorch-poetry-generation/blob/master/word_language_model/GENERATED/Terminal Saved Output-19042017-1331-1MCorpus-2h48m.txt.

[126] Adam Downer, "Harry Potter Predictive Text Chapter," Know Your Meme, January 25, 2018. https://knowyourmeme.com/memes/harry-potter-predictive-text-chapter.

[127] Pold and Erslev, *Data-Realism*, 2020.

definition of a line was in order to create her poetic output. Johnston's definition of a poem, which enables the generation, is a sequence of 88 words, broken up into lines in various ways.



```
for i in range(args.words):
    output, hidden = model(input, hidden)
    word_weights = output.squeeze().data.div(args.temperature).exp().cpu()
    word_idx = torch.multinomial(word_weights, 1)[0]
    input.data.fill_(word_idx)

    if word_idx<=len(corpus.dictionary.idx2word)-1:
        word = corpus.dictionary.idx2word[word_idx]

        if word == '<eos>':
            word = '\n'

        if word == '&amp;':
            word = '\n'



        words+=word+" "
```

Figure 13. *A Section of the generation code that looks up output words based on their indices and concatenates them into an 88-word poem.*

For each poetic chunk that is generated by the PyTorch model, the first line is designated as the title. If that line is not empty, each word is then capitalized and another blank new line is added between the title and the poem. This adds a significant amount of whitespace that is also part of the textual output and changes the material that is available to "carve." Throughout Johnston's many different models over the course of the year, the structure of this output changes as well. Sometimes a poem takes the form of lineated verse with a title, sometimes it takes the form of a small paragraph. This parsing and formatting of the output changed the shape and material of the textual output that was then carved.

Pagination

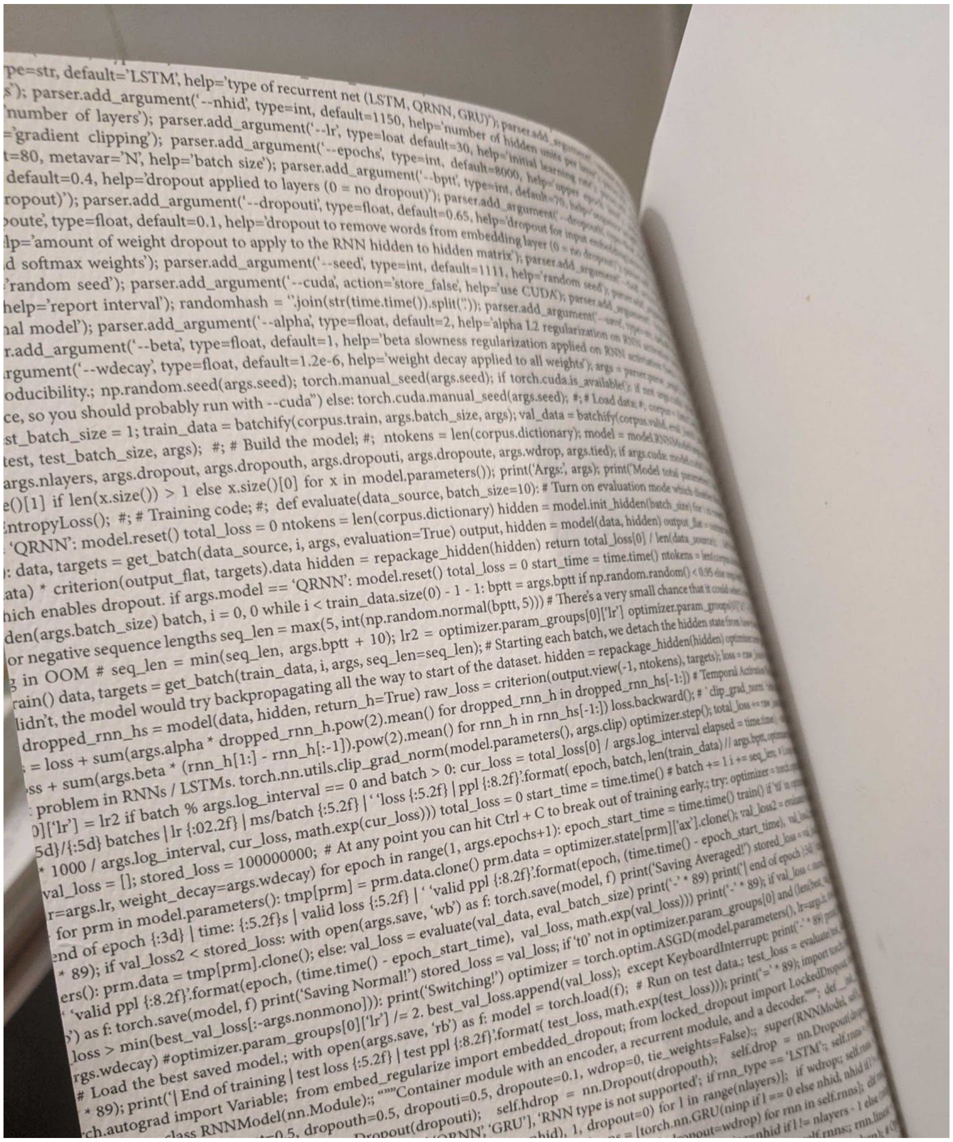
ReRites - Raw Output + Responses is not only a sort of summarization of the twelve-book box set, but also of performances and exhibitions of the work that spawned from the project. There is a strong performative streak in the writing process of the project as well. Later in Johnston's year of writing, he began to record the "carving" process in order to perform the editing and also use surveillance as a writing technique. He began to use this technique of recording to motivate other writing projects as well. Even if no one was watching, he would be motivated to continue writing because of the feeling or expectations of a performance. The videos of the carving create a sense of drama and demonstrate how personal the editorial process is. When you watch Johnston's cursor as he carves, you are forced to watch the potential of the text go in the direction of Johnston's decision-making. Huge chunks of texts are deleted and it is sometimes shocking or saddening to see some beautiful lines erased. Watching the carving gives you a sense of how a different author would create a very different collection of poetry and how the lengthy duration of Johnston's experiment affected the way he dealt with the "raw output." Due to the many hours that Johnston put into editing, the volume of work, and the ritualistic nature of the practice, he eliminates too much time for contemplation, and instead just moves with the momentum of what he calls the, "gravitational path" of "cutting and carving."[128]

Unlike *Articulations* and *1 the Road*, which both use conventional bookish serif fonts in their final publication, the poems in *ReRites - Raw Output + Responses,* along with all of the contextualizing essays and materials, are published in the same Ubuntu font that can be seen in the videos of Johnston's carving in a Sublime text editor.[129] The front and back end pages of the book, however, are decorated with Johnston's code, sized very small and changed to a serif font,

[128] Johnston, Appendix C.

[129] Johnston, *RERITES*. http://glia.ca/rerites/.

which frames the text at each end of the codex. This is the first page that a reader sees if they were to open the book sequentially. The code here is not meant to be legible or to provide full documentation of any part of the project, but rather cuts off at the margins and fills the page as a sort of illustration.



```
pe=str, default='LSTM', help='type of recurrent net (LSTM, QRNN, GRU)'); parser.add_
s'); parser.add_argument('--nhid', type=int, default=1150, help='number of hidden units per
'number of layers'); parser.add_argument('--lr', type=loat default=30, help='initial learning rate'
='gradient clipping'); parser.add_argument('--epochs', type=int, default=8000, help='upper
t=80, metavar='N', help='batch size'); parser.add_argument('--bptt', type=int, default=70, help=
 default=0.4, help='dropout applied to layers (0 = no dropout)'); parser.add_argument('--dropout
ropout)'); parser.add_argument('--dropouti', type=float, default=0.65, help='dropout for input
oute', type=float, default=0.1, help='dropout to remove words from embedding layer (0 = no dropout)
lp='amount of weight dropout to apply to the RNN hidden to hidden matrix'); parser.add_argument
d softmax weights'); parser.add_argument('--seed', type=int, default=1111, help='random seed');
'random seed'); parser.add_argument('--cuda', action='store_false', help='use CUDA'); parser.add_
help='report interval'); randomhash = ''.join(str(time.time()).split('.')); parser.add_argument('--save
al model'); parser.add_argument('--alpha', type=float, default=2, help='alpha L2 regularization on RNN
r.add_argument('--beta', type=float, default=1, help='beta slowness regularization applied on RNN activation
rgument('--wdecay', type=float, default=1.2e-6, help='weight decay applied to all weights'); args = parser.parse
oducibility.; np.random.seed(args.seed); torch.manual_seed(args.seed); if torch.cuda.is_available(): if not args.cuda
ce, so you should probably run with --cuda") else: torch.cuda.manual_seed(args.seed); #; # Load data; #, corpus =
st_batch_size = 1; train_data = batchify(corpus.train, args.batch_size, args); val_data = batchify(corpus.valid, eval_
test, test_batch_size, args);  #; # Build the model; #;  ntokens = len(corpus.dictionary); model = model.RNNModel
args.nlayers, args.dropout, args.dropouth, args.dropouti, args.dropoute, args.wdrop, args.tied); if args.cuda: model
e()[1] if len(x.size()) > 1 else x.size()[0] for x in model.parameters()); print('Args:', args); print('Model total param
ntropyLoss();  #; # Training code; #;  def evaluate(data_source, batch_size=10): # Turn on evaluation mode which dis
 'QRNN': model.reset() total_loss = 0 ntokens = len(corpus.dictionary) hidden = model.init_hidden(batch_size) for i
: data, targets = get_batch(data_source, i, args, evaluation=True) output, hidden = model(data, hidden) output_flat =
ata) * criterion(output_flat, targets).data hidden = repackage_hidden(hidden) return total_loss[0] / len(data_source)
hich enables dropout. if args.model == 'QRNN': model.reset() total_loss = 0 start_time = time.time() ntokens = len(corp
den(args.batch_size) batch, i = 0, 0 while i < train_data.size(0) - 1 - 1: bptt = args.bptt if np.random.random() < 0.95 else
or negative sequence lengths seq_len = max(5, int(np.random.normal(bptt, 5))) # There's a very small chance that it could
g in OOM # seq_len = min(seq_len, args.bptt + 10); lr2 = optimizer.param_groups[0]['lr'] optimizer.param_groups[0]
rain() data, targets = get_batch(train_data, i, args, seq_len=seq_len); # Starting each batch, we detach the hidden state from
lidn't, the model would try backpropagating all the way to start of the dataset. hidden = repackage_hidden(hidden) optimizer
 dropped_rnn_hs = model(data, hidden, return_h=True) raw_loss = criterion(output.view(-1, ntokens), targets); loss = raw_
 = loss + sum(args.alpha * dropped_rnn_h.pow(2).mean() for dropped_rnn_h in dropped_rnn_hs[-1:]) # Temporal Activation
ss + sum(args.beta * (rnn_h[1:] - rnn_h[:-1]).pow(2).mean() for rnn_h in rnn_hs[-1:]) loss.backward(); # `clip_grad_norm`
 problem in RNNs / LSTMs. torch.nn.utils.clip_grad_norm(model.parameters(), args.clip) optimizer.step(); total_loss += raw_
0]['lr'] = lr2 if batch % args.log_interval == 0 and batch > 0: cur_loss = total_loss[0] / args.log_interval elapsed = time.time
5d}/{:5d} batches | lr {:02.2f} | ms/batch {:5.2f} | ' 'loss {:5.2f} | ppl {:8.2f}'.format( epoch, batch, len(train_data) // args.bptt, optim
* 1000 / args.log_interval, cur_loss, math.exp(cur_loss))) total_loss = 0 start_time = time.time() # batch += 1 i += seq_len; #
val_loss = []; stored_loss = 100000000; # At any point you can hit Ctrl + C to break out of training early.; try: optimizer = torc
r=args.lr, weight_decay=args.wdecay) for epoch in range(1, args.epochs+1): epoch_start_time = time.time() train() if 't0'
 for prm in model.parameters(): tmp[prm] = prm.data.clone() prm.data = optimizer.state[prm]['ax'].clone(); val_loss2 =
end of epoch {:3d} | time: {:5.2f}s | valid loss {:5.2f} | ' 'valid ppl {:8.2f}'.format(epoch, (time.time() - epoch_start_time),
* 89); if val_loss2 < stored_loss: with open(args.save, 'wb') as f: torch.save(model, f) print('Saving Averaged!') stored_
ers(): prm.data = tmp[prm].clone(); else: val_loss = evaluate(val_data, eval_batch_size) print('-' * 89) print('| end of ep
' 'valid ppl {:8.2f}'.format(epoch, (time.time() - epoch_start_time),  val_loss, math.exp(val_loss))) print('-' * 89); if va
') as f: torch.save(model, f) print('Saving Normal!') stored_loss = val_loss; if 't0' not in optimizer.param_groups[0] an
loss > min(best_val_loss[:-args.nonmono])): print('Switching!') optimizer = torch.optim.ASGD(model.parameters
rgs.wdecay) #optimizer.param_groups[0]['lr'] /= 2. best_val_loss.append(val_loss);  except KeyboardInterrupt: pri
# Load the best saved model.; with open(args.save, 'rb') as f: model = torch.load(f);  # Run on test data.; test_loss
 * 89); print('| End of training | test loss {:5.2f} | test ppl {:8.2f}'.format( test_loss, math.exp(test_loss))); print('=' * 8
ch.autograd import Variable;  from embed_regularize import embedded_dropout; from locked_dropout import
 class RNNModel(nn.Module):; """Container module with an encoder, a recurrent module, and a dec
 =0.5, dropouth=0.5, dropouti=0.5, dropoute=0.1, wdrop=0, tie_weights=False):;  supe
 Dropout(dropouti);   self.hdrop  =  nn.Dropout(dropouth);   self.drop =
 'QRNN', 'GRU'], 'RNN type is not supported'; if rnn_type ==
 hid), 1, dropout=0) for l in range(nlayers)]
 = [torch.nn.GRU(ninp if l ==
 dropout=wdrop) for r
 nhid i
```

Figure 14. Inside front cover of *ReRites - Raw Output + Responses*

For *ReRites*, the bulk of the writerly labor happens during the pagination process, which includes editing. While Johnston's process of carving occurred over the course of a year and resulted in the twelve book-length works in the box set, selecting which poems would appear in the smaller commercial version was another phase in the editorial process. Johnston's selections were based on personal preference and also on how each poem would represent the project as a whole.[130] The pagination stage also included soliciting responses from colleagues and other "sophisticated literary/digital practitioners,"[131] including Allison Parrish, Johanna Drucker, Kyle Booten, John Cayley, Lai-Tze Fan, Nick Montfort, Mairéad Byrne, and Chris Funkhouser.

[130] Johnston, Appendix C.

[131] Johnston, *ReRites - Raw Output + Responses*, 99.

3. WAYS OF WRITING

WRITERLY PRACTICES

While each of these individual authorial decisions may not be entirely unique to AI-generated literature, they are all specific areas of potential writerly intervention in the development of an AI text-generation system. While these might apply to multiple computational text-generation techniques and to various mediums, these are some of the writerly practices that are collectively unique to AI-generated books. It is useful to categorize these authorial techniques using Espen Aarseth's typology of preprocessing, coprocessing, and postprocessing for ease of categorization and to enable discussions about particular authorial practices that emphasize each of these aspects of computational writing.[132]

Preprocessing

We have seen that *Articulations*, *1 the Road*, and *ReRites* all involved the creation of unique datasets that affected the textual output of each respective system. The preprocessing work for *Articulations* involved the development and documentation of "A Gutenberg Poetry Corpus" wherein millions of lines of public domain poetry from Project Gutenberg were collected and categorized for creative use. *1 the Road* involved the selection of certain texts, largely somber Eastern European novels, that could statistically produce an intentionally bleak tone. However, Goodwin used multiple trained models for the generation of *1 the Road* and part of his practice involves constantly creating and maintaining different collections of texts that can produce different types of literary output when used as a training corpus. Similarly, the dataset for *ReRites* included twentieth century avant-garde poetry scraped from the web, song lyrics

[132] Aarseth, *Cybertext*, 135.

from the likes of Tom Waits and Patti Smith, research articles to provide the system with vocabulary surrounding relevant topics such as posthumanism, excerpts from news articles and blogs, a collection of quotes that were archived by Johnston, and, as the year of work progressed, previous months of *ReRites* poetry. The creation of a training corpus is a writerly process mandated by AI-generated literature that opens up many avenues for interpretation. Even if the training corpus is not selected by an individual author/programmer, but rather developed by researchers as part of a publically available pre-trained model such as GPT-2, the training data, where it comes from, and how it is sourced from various networked publics are all still relevant areas of analysis. GPT-2 was trained on a dataset of outbound links from Reddit receiving at least 3 karma.[133] Google's BERT was trained on the (mostly) human-generated text of Wikipedia and the Google Books corpus.[134]

The development and training of a machine learning system is also not a trivial amount of labor. It is a process that involves the selection of certain platforms, from programming languages and modules to the selection of certain hardware like laptops, CPUs, or GPUs, which all have broader implications for the work. A machine learning model also needs to be developed or chosen and formatted. The programming done for *Articulations* involved a novel method for embedding textual data in a phonetic vector space, *1 the Road* utilized the publicly available torch-rnn LSTM model, while *ReRites* used a range of available models that became available throughout the year that the project was written. Additionally, if applicable, other parts of the system need to be automated as well. Allison Parrish had to develop heuristic methods for

[133] Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. "Better Language Models and Their Implications," OpenAI, February 14, 2019. https://openai.com/blog/better-language-models/.

[134] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," May 24, 2019. https://arxiv.org/pdf/1810.04805.pdf.

programming paragraph and line breaks in *Articulations*, as well as the random walk. Ross Goodwin developed surrounding systems to automatically incorporate the GPS, time, and image data coming into the system. The selection of these sources of data is also part of the preprocessing procedure, as ways of sensing and modes of data collection will contextualize and inform any text-generation system.

Coprocessing

The generation process for *1 the Road* involved switching between trained models depending on the author/programmer's literary and aesthetic preferences. For the project, Goodwin used three trained models that he was relatively familiar with, one trained on bleak Eastern European novels, one trained on science fiction prose and poetry, and the last trained on poetry from around the world. Throughout the road trip, Goodwin would change which trained model the generated text was coming from a few times throughout each day, whenever he felt the text was getting too monotonous. Goodwin didn't constantly change the models, but oversaw the direction of the larger system. Another form of guidance and coprocessing that plays a significant part in *1 the Road* is the human direction of the associated sensors, the driving of the car and GPS unit, the placement of the camera, the interaction with the microphone and the locative specifics of the time data. The entire road trip and generation process is one of coprocessing.

Postprocessing

For any non-deterministic text-generation system, and often for deterministic systems that produce outputs too extensive to reasonably fit into a printed book, a significant part of an author/programmer's postprocessing workflow will be to either run a text-generation system many times until a favorite textual output is chosen or to select certain passages that are more entertaining or that are representative of what the larger work is trying to achieve. For *Articulations*, Allison Parrish ran the random walk through the vector space multiple times to generate around 20 different drafts of the book. She ultimately chose the version that she thought had the most interesting beginning and ending. This editorial choice is interesting, especially in the context of the program used to generate the text and what the possibility space is in the broader generating potential of the system. The selection of an output like this is reflective of certain poetic ideals and intentions that the author/programmer may carry. This editorial process of selection is part of previous accusations of trickery when it comes to AI-generated text. Controversy surrounding *The Policeman's Beard is Half-Constructed* and its AI author RACTER focused on the amount of editorial and creative labor that the author/programmer behind the work, William Chamberlain, was responsible for. *The Policeman's Beard* is a collection of RACTER's carefully-selected gems, and this type of selection was used to argue that the text was not actually AI-generated.[135] However, we have seen that human labor must be involved in any type of text-generating process and that the ways that different systems incorporate human labor exemplify certain types of writerly practices. *ReRites*, the smaller commercial book that was designed to represent the broader year-long, twelve-volume project also features this type of selection. 60 poems were selected from all twelve volumes of poetry to provide a sense of the larger work.

Another type of postprocessing is the removal or deletion of parts of the generated output. One line of the generated prose poem of *Articulations* was removed after the author came

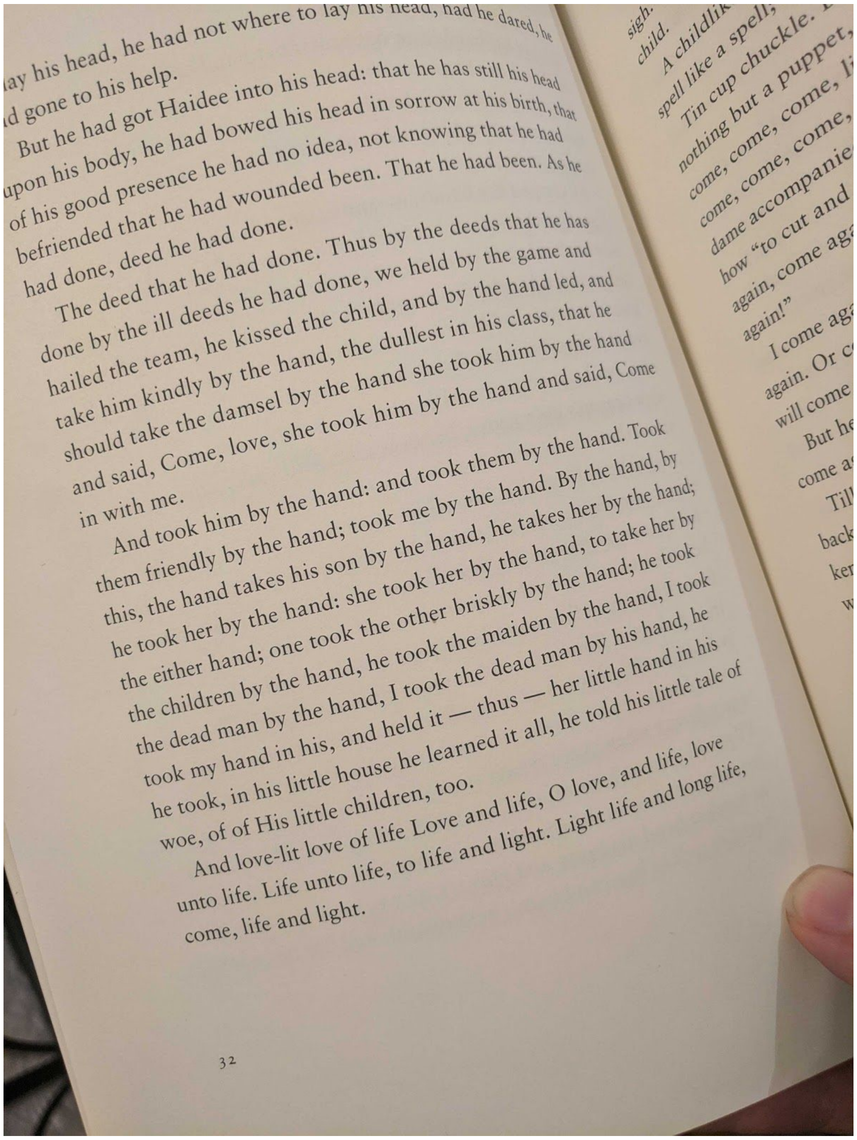
[135] Aarseth, *Cybertext*, 132.

across a word that she didn’t feel comfortable having in herpiece. The output of *1 the Road* also went through a similar editorial process. The project was built to be intentionally sparse so that any publisher would not be able to edit the work down into something conventionally high-quality, and to keep the work focused on the raw output that was generated during the road trip. While editorial compromises were still made, they were minimal and purely subtractive. This process is not new, and is a kind of extension of the tradition of erasure poetry where poems are created by erasing words in a source text. Erasure poetry also follows the same thread of potentiality and latency that machine learning explores, and exemplifies the idea that poetic relationships lie latent within existing corpora waiting to be discovered. While the subtractive editorial processes here are not meant to be the driving force of the text-generation systems, they speak to certain necessities of printing this kind of work in the form of a book and illustrate ways that author/programmers who are invested in the absence of postprocessing in their work make compromises. David Jhave Johnston’s editorial practice was also one largely of erasure. He called the process of waking up every morning and editing AI-generated text for two hours one of “carving.” He did not edit and reword in the traditional sense, but chipped away at large amounts of output to carve out poems from the “raw” text. He did add some punctuation and capitalization, but the vocabulary all belonged to the system. Still, the output was extensive enough and the vocabulary large enough to offer a lot of flexibility in what could be said or built with the textual material.

DIVERGENT THREADS OF PRACTICE

Both Allison Parrish and Ross Goodwin aim to minimize, if not eliminate, any sort of postprocessing poetic decisions, whereas David Jhave Johnston explores machine learning specifically with an emphasis on postprocessing and heavy editorial decision making, describing the project as a process of programming a virtual muse"[136] in reference to Charles O. Hartman's famous book of the same title.[137] *Articulations*, *1 the Road*, and *ReRites* are each examples of emphasizing one of the three categories of preprocessing, coprocessing, and postprocessing. For Allison Parrish, the primary poetic operation of machine learning is the exploration of a vector space, or the ability to traverse and sometimes generate items that are similar to each other. This particular authorial stance focuses on the poetic decisions made during the preprocessing of a text-generation system and keeps coprocessing or postprocessing to a minimum.

[136] Johnston, *ReRites - Raw Output / Responses*, 176.
[137] Charles O. Hartman, *Virtual Muse: Experiments in Computer Poetry* (Wesleyan University Press, 1996).



ay his head, he had not where to lay his head, had he dared, he
d gone to his help.

But he had got Haidee into his head: that he has still his head upon his body, he had bowed his head in sorrow at his birth, that of his good presence he had no idea, not knowing that he had befriended that he had wounded been. That he had been. As he had done, deed he had done.

The deed that he had done. Thus by the deeds that he has done by the ill deeds he had done, we held by the game and hailed the team, he kissed the child, and by the hand led, and take him kindly by the hand, the dullest in his class, that he should take the damsel by the hand she took him by the hand and said, Come, love, she took him by the hand and said, Come in with me.

And took him by the hand: and took them by the hand. Took them friendly by the hand; took me by the hand. By the hand, by this, the hand takes his son by the hand, he takes her by the hand; he took her by the hand: she took her by the hand, to take her by the either hand; one took the other briskly by the hand; he took the children by the hand, he took the maiden by the hand, I took the dead man by the hand, I took the dead man by his hand, he took my hand in his, and held it — thus — her little hand in his he took, in his little house he learned it all, he told his little tale of woe, of of His little children, too.

And love-lit love of life Love and life, O love, and life, love unto life. Life unto life, to life and light. Light life and long life, come, life and light.

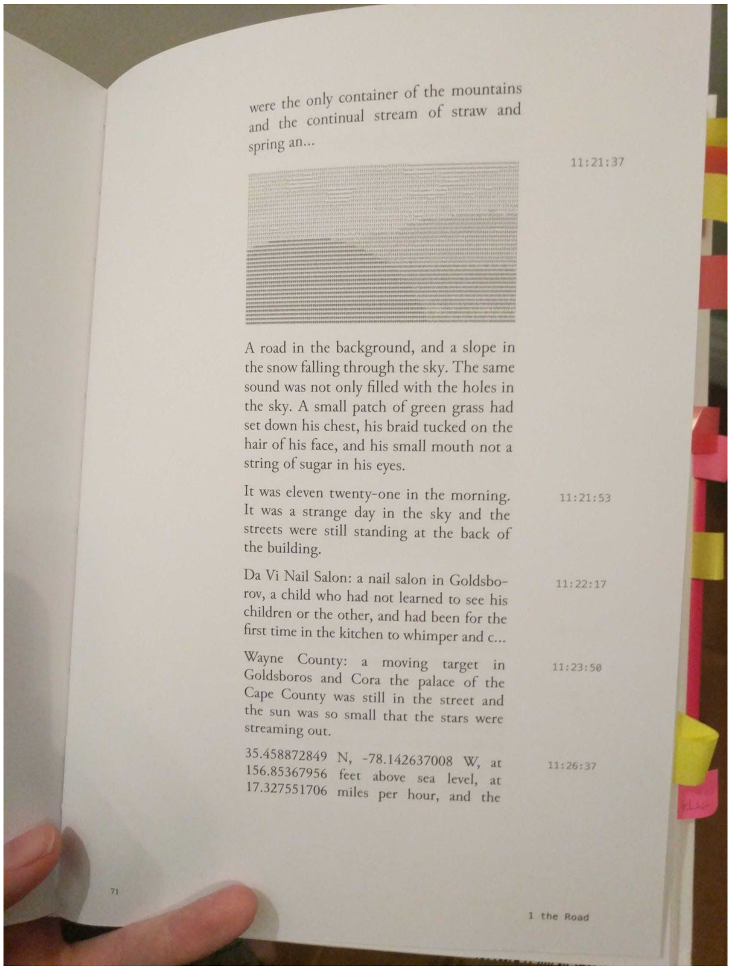
32

sigh.
child.
A childlik
spell like a spell;
Tin cup chuckle.
nothing but a puppet,
come, come, come, li
come, come, come,
dame accompanie
how "to cut and
again, come ag
again!"
I come ag
again. Or c
will come
But he
come a
Till
back
ker
w

Figure 15. *Excerpt from* Articulations *that represents the poetic gradient that comes from traversing a vector space.*

The passage in Figure 15 is a representative example of this kind of traversal. "Come, love, she took him by the hand and said, Come in with me," is the beginning of a section that explores the lines of poetry in the English language that are related to taking someone's hand. The phonetic relationships, applied to the unit of the line, also act as a sort of context window, creating these topic-focused passages that also create a type of emergent narrative, not only through the collective voice of poets regarding this topic, but in the accumulation of lines that creates an aesthetic and narrative experience.

1 the Road emphasizes the creation of writing interfaces that are able to be utilized through coprocessing. The project as a whole turns the surveillance vehicle into an interface for writing, with each mode of surveillance adding a feature to the "car as a pen." This carries through to Goodwin's other work, such as word.camera and Please Feed the Lions.



were the only container of the mountains and the continual stream of straw and spring an...

11:21:37

A road in the background, and a slope in the snow falling through the sky. The same sound was not only filled with the holes in the sky. A small patch of green grass had set down his chest, his braid tucked on the hair of his face, and his small mouth not a string of sugar in his eyes.

It was eleven twenty-one in the morning. 11:21:53
It was a strange day in the sky and the streets were still standing at the back of the building.

Da Vi Nail Salon: a nail salon in Goldsbo- 11:22:17
rov, a child who had not learned to see his children or the other, and had been for the first time in the kitchen to whimper and c...

Wayne County: a moving target in 11:23:50
Goldsboros and Cora the palace of the Cape County was still in the street and the sun was so small that the stars were streaming out.

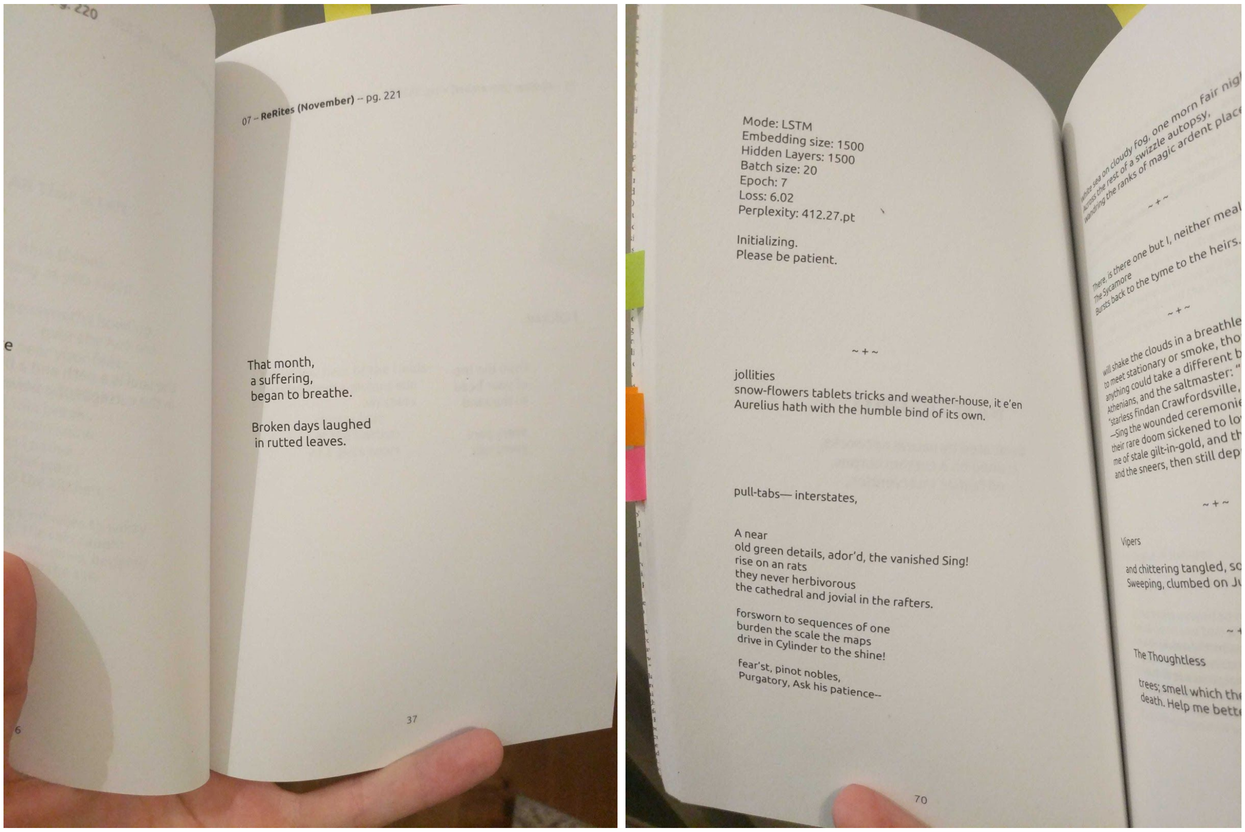
35.458872849 N, -78.142637008 W, at 11:26:37
156.85367956 feet above sea level, at 17.327551706 miles per hour, and the

71

1 the Road

Figure 16. *Excerpt from* 1 the Road *that features entries that use every type of sensory data collected by the surveillance car system*

The text of *1 the Road* is a log of coprocessing efforts that all seed into the neural text-generation system in some way. As the car is driven from New York to New Orleans, all of the sensors are guided by human efforts, and the models into which the collected data are seeded are also controlled by the human collaborator. Although *ReRites* also focuses on a type of collaboration, it does not emphasize coprocessing. Rather, the collaborative editorial process occurs in the realm of postprocessing. A massive amount of output is generated, and after the fact, Johnston "carved" the material to create poems.



07 -- **ReRites (November)** -- pg. 221

That month,
a suffering,
began to breathe.

Broken days laughed
in rutted leaves.

37

```
Mode: LSTM
Embedding size: 1500
Hidden Layers: 1500
Batch size: 20
Epoch: 7
Loss: 6.02
Perplexity: 412.27.pt

Initializing.
Please be patient.
```

~ + ~

jollities
snow-flowers tablets tricks and weather-house, it e'en
Aurelius hath with the humble bind of its own.

pull-tabs— interstates,

A near
old green details, ador'd, the vanished Sing!
rise on an rats
they never herbivorous
the cathedral and jovial in the rafters.

forsworn to sequences of one
burden the scale the maps
drive in Cylinder to the shine!

fear'st, pinot nobles,
Purgatory, Ask his patience--

70

White sea on cloudy fog, one morn fair nig
Across the rest of a swizzle autopsy,
Wandring the ranks of magic ardent plac

~ + ~

There, is there one but I, neither meal
The Sycamore
Bursts back to the tyme to the heirs.

~ + ~

will shake the clouds in a breathle
to meet stationary or smoke, tho
anything could take a different b
Athenians, and the saltmaster: "
"starless Findan Crawfordsville,
—Sing the wounded ceremoni
their rare doom sickened to lo
me of stale gilt-in-gold, and th
and the sneers, then still dep

~ + ~

Vipers

and chittering tangled, sc
Sweeping, climbed on Ju

The Thoughtless

trees; smell which the
death. Help me bett

Figure 17. *(Left): a "carved" poem from ReRites generated in November 2017, (Right): "Raw output"*

Even though all of these authors know, or know of each other, and are in a community of practice of people experimenting with computational writing, the ecosystem of AI-generated literature does not make up a movement with a manifesto or a particular aesthetic or agenda, but reflects a range of emerging authorial practices that diverge and sometimes conflict with each other. Johnston's programming of a "virtual muse" is a different perspective of AI that leads to artistic practices and explorations that diverge from Goowin's conceptualization of Narrated Reality and being a "writer of writers." Both of these perspectives differ from Parrish's programmatic poetics that emphasizes the creation of novel computational models. From comparing these three writers and the way that they conceptualize their practices, Henrickson's continuum of generatorship to authorship applies very well in terms of categorizing the agency of the machine. The closer the author/programmer is to the computational inner-workings of the machine learning model, the more authorial credit is taken, while the projects that emphasize coprocessing and postprocessing slowly begin to introduce intermediary authorial definitions like "writer of writer," and "virtual muses."

Another way of writing that is unique to each author/programmer is the writing of the code itself. Every programmer has their own style of coding, and this is a type of writing as well. [138] Not only the writing of the code, but the way that it is maintained and stored also points to specific authorial practices and relationships to computing cultures. Parrish, as a former software developer and current teacher of creative computing, emphasizes the thorough documentation and the public accessibility of her projects for students. This, as is evident in the amount of detail that I was able to explore her process, is very helpful for researchers in some ways. Goodwin's repositories very much reflect the threads of his ideas and the development of different creative directions. Goodwin's programming journey has centered around his creative ambitions and his code often describes the hardware and specifications of the interfaces very clearly while repurposing code to ultimately achieve an experiential output rather than a programmatic one. Similarly, Johnston used his *ReRites* repositories as an experimental lab rather than a neatly

[138] Angus Croll, *If Hemingway Wrote JavaScript* (No Starch Press, 2014).

edited final system. In his code you can find early iterations of writing with different neural networks, many different training datasets and generated samples, and documentation of extensive experimentation.

4. WAYS OF READING

Along with a diversity of authorial and writerly practices in AI-generated literature, there are a wealth of approaches for reading this kind of text. The material analysis of the book, the interpretation of paratextual elements, and the exploration of authorial intention through the code and writerly decisions made in the development of the underlying programs are some of the approaches for reading AI and computer-generated texts more broadly. In a survey of reader-responses to text that has been generated by an NLG system, Leah Henrickson writes that readers are not certain how to read computer-generated texts and can potentially react in a hostile way to the unfamiliar and perhaps new relationships between writer, reader, and text that arise from "algorithmic authorship."[139] In a focus group of readers, the majority of responders attributed authorship of a sample of computer-generated text to the program, choosing among a range of responses including the human programmer who worked with the system, the corporation that owns the program, and the funders of said corporation. This attribution of authorial credit to the machine can obfuscate other entry points into ways of reading and analyzing AI-generated texts. Additionally, the label of AI, rather than just computer-generated, can add another layer of obfuscation due to sometimes over-exaggerated associations of AI with human-level intelligence or consciousness. If one perceived AI in this way, and tried to read the output of AI-generated literature only as the intelligent expression of a machine, many important elements of the text will be overlooked. While the text itself is a crucial part of the reading experience of AI-generated text, any textual analysis should be informed by material, paratextual, and intentional readings.

[139] Henrickson, *Towards a New Sociology of the Text*, 177.

MATERIAL, PARATEXTUAL, INTENTIONAL, TEXTUAL

The physical book, materially and traditionally typeset, sets up particular expectations about operations of reading and writing through the lens of print-centric literary studies. However, printed books today are digital objects. They are typed up, designed on digital platforms, printed with links to publisher's homepages and author Twitter accounts and Library of Congress permalinks on their end pages. They are born of and materially embedded in networked culture while new media technologies reframe their affordances. Jessica Pressman uses the book House of Leaves as an example of what she calls the "networked novel,"[140] which reflects and refracts digital ecosystems and aspects of networked culture in print. She suggests that this kind of work also fosters new types of reading that are distributed across multimedia networks. The way that electronic media changes reading, writing, and the printed book does not only apply to novels and relate to networks, but as Katherine Hayles notes, it more broadly applies to works of literature and relates to computation. In Hayles's proposed dynamic heterarchies of intermediation, works of literature are affected by computation at every instance of interaction between agents in any assemblage of labor:

> Literature, conceptualized not just as print books but as the entire complex system of literary production that includes writers, editors, publishers, critics, designers, programmers, booksellers, readers, players, teachers, copyright laws and other legal formations, Web sites and other electronic dissemination mechanisms, and the technologies that enable and instantiate all of the above, is permeated at every level by computation.[141]

This points to the fact that producing a book is not entirely the effort or work of an

[140] Jessica Pressman, "House of Leaves : Reading the Networked Novel," *Studies in American Fiction* 34, No. 1 (2006), 108. https://doi.org/10.1353/saf.2006.0015.
[141] Hayles, *Intermediation*, 122.

isolated individual, regardless of any sort of computational intervention in the writing process. It also clearly positions the printed book as an object that is mediated by computation. There are many works of electronic literature that, as the term's definition calls for, could not have been produced without the capabilities and contexts of the standalone or networked computer, and yet are displayed and consumed in the form of a printed codex book. These include novels that take the form of emails, SMS messages, or blogs, but also works of literature that have been computationally generated in some way and formatted for print. AI-generated literature in print points to specific practices and material relationships that do not merely retrofit computationally-generated text to an obsolete medium, but consciously use the printed book as the display for a text that is somehow dependent on the affordances of computation. While many creative technologists and e-lit authors focus on more high-tech or multimedia displays, others turn to the format of the printed book. The influence of computation on social, cultural, and manufacturing processes influence the material book, but the inscribed text itself is also affected by the prevalence of digital media. In *Aesthetic Animism*, Johnston writes that ubiquitous computation marks a "change in the ontological status of language from abstraction to entity to ecological force."[142] In *Articulations*, *1 the Road*, and *ReRites*, the printed text is made up of concrete units of language that exist as entities that have been categorized, organized, and manipulated.

Material investigation of this data can lead to interesting discoveries about cultures of information, the creation and storage of a text, about the manipulation of that text by the author for poetic purposes and also opens up a range of "readings" of the associated datasets, code,

[142] Johnston, *Aesthetic Animism*, 11.

algorithms, and platforms. Additionally it is possible in some cases, as with *Articulations*, to identify the source texts and include the original sources in the reading of the work. This is also not a new method of reading, and can be used even for Burroughs' analog sample of his proposed cut-up method: "Visit of memories. Only your dance and your voice house. On the suburban air improbable desertions ... all harmonic pine for strife."[143]This section of text is a cut-up of Arthur Rimbaud's poem *Youth*.[144] This type of print-based hypertext is one way that the philosophies surrounding computationally mediated text become incorporated into the printed book. When it is possible to identify parts of a source text in the generated output, this can provide additional areas of analysis, but ultimately something new is created. John Cayley's *How It Is In Common Tongues* pushed this premise to its limit by printing the complete text of Samuel Beckett's *How It Is*, but with chunks of the entire text located on the web and cited so that the entire text no longer belongs to Beckett, but links out into a hypertextual network of universally accessible language.

Many noteworthy features of the material book fall under the category of what Gérard Genette has coined *paratexts*, or "liminal devices" within and without the physical book itself that mediate the relationship between texts, authors, publishers, and readers.[145] While the term paratext typically takes on a broader meaning in the field of games studies, a stricter application of Genette's original term is useful for the analysis of AI-generated literature in the form of a printed book. This includes elements that can be found on the book itself like the price, the author attribution, introductions and other contextualizing essays, the library of congress information, publisher websites, and more. The bright pink band around *1 the Road*'s otherwise

[143] Montfort, "The Cut-Up Method," 2003.

[144] Arthur Rimbaud, "Youth." PoemHunter.com, April 3, 2010. https://www.poemhunter.com/poem/youth-5/.

[145] Gérard Genette, *Paratexts: Thresholds of Interpretation*. Cambridge University Press, 1997.

plain white cover is one example of a paratextual feature that the publisher uses to communicate with readers. The band, which is a sort of mini dust jacket that is frequently found in French publishing, is typically a strip of brightly colored paper that encases about one third of the book. Genette refers to this band as a material "appendage of the cover."[146] Because it is something that can be easily removed from the book, he notes that any text on the band is likely to be transitory in nature and something to be forgotten after making its impression.[147] *1 the Road*'s pink band advertises the book by stating that "The first gonzo Artificial Neural Network is a genius writer." This is the claim of first-ness that Goodwin was able to negotiate onto the temporary sleeve. The publisher wanted to make this sort of impression for sales, and Goodwin wanted to make sure that the book itself did not claim novelty, leaving the liminal space of the band as a perfect compromise.

Additional inconsistencies in marketing emerge from the categorization of *1 the Road*. While the press release clearly states that it is a novel, one of the introductory essays by Kenric McDowell, the head of the Artist + Machine Intelligence program at Google Arts & Culture, refers to the project as "AI poetry."[148] This essay is accompanied by an essay of Goodwin's own, which refers to the text only as a "manuscript" or "book," resisting poetic or novelistic classification. *1 the Road*, *Articulations*, and *ReRites* all include some sort of accompanying introduction or essay written by the author that addresses the methods of text generation. This not only suggests that author/programmers think that an understanding of the computation behind the work informs the reading in a crucial way, but also justifies the author attribution in making clear the amount of human labor that went into the process of writing. Parrish's

[146] Genette, *Paratexts*, 27.
[147] Ibid., 28.
[148] Goodwin, *1 the Road*, 18.

introduction is minimal and focuses only on explaining the computational methods behind the generation of *Articulations*. Goodwin's introduction briefly touches on method but focuses more on his philosophy of AI as a sort of "camera for writing." Johnston's "Human-Author End-Note" summarizes his year of ritualistic practice and computational techniques and also claims a first of its own: "the first sustained use of deep learning as virtual muse."[149]

The paratextual material does not only justify to a readership that the text printed within is approachable and worthy of attention, but also justifies the work to scholars in what Johnston describes as an, "attempt to bridge the credibility gap," for digital poetry within literary communities. He says that, "in many people's minds the kind of poetry that computers are writing, even when we use a hybrid-augmented technique, it's insufficient." Johnston reflects that there is a sense that computer-generated literature is "tainted or contaminated by the influx of the machine," and that any sort of computational augmentation must reflect some sort of insufficiency on the part of the human writer. This need to justify the work is a fight for recognition and respect. *ReRites* includes the accompanying responses by scholars and creators of digital media art in order to provide a kind of endorsement and introduce the work of digital poetry to a more traditional literary audience.

The novelty of computer-generated and AI-generated literature makes paratextual posturing important to this type of work. However, paratexts are not necessarily more important in the reading of computer-generated works than traditional literature. Johnston says that for most computer-generated literary art, he could, "watch it long enough to know kind of what it's doing behind the scenes."[150] Reading just the text to look for underlying algorithmic processes,

[149] Johnston, *ReRites - Raw Output / Responses*, 176.
[150] Johnston, Appendix C.

rather than being briefed on the mechanisms of generation ahead of time, is also a way of reading this type of work. In 2016, Johnston speculated that the future of poetic readings would be to "correlate manually chosen features (like line length, word length, vocabulary, metrical regularity, length of verse, etc.) with the results of the automated unsupervised learning algorithms."[151] Kyle Booten and Dan Rockmore write that computer-generated literary text is usually not developed to seamlessly mimic human communication, but rather to textualize and lay bare a system's "algorithmic bones."[152] While the algorithms themselves are not a replacement for the text, the code can be a part of the reading, especially as a metric for authorial intent. The author/programmer of a literary text-generation system has to intentionally add every part of the program, even if this intentionality is randomness and obfuscation. An intentional reading of the work should also pay close attention to any interpretive direction provided by the author in the contextualizing materials such as introductions or author biographies. Parrish's introduction signals the importance of the development of the model, while elements like Goodwin's claim that he is "not a poet," can provide additional commentary next to an essay describing his work as "machine poetry."

Lastly, the text can be read for more conventional readerly experiences like narrative. Narrative, defined as any sequential representation of events that consists of actions (story) and representations of those actions (narrative discourse) applies to *Articulations*, *1 the Road*, and *ReRites - Raw Output + Responses*.[153] The main narratives, or sequences of events, in each are the random walk, the road trip, and the year of carving, respectively.

[151] Johnston, *Aesthetic Animism*, 127

[152] Kyle Booten and Dan Rockmore, "The Anxiety of Imitation: On the 'Boringness' of Creative Turing Tests," electronic book review, May 3, 2020. https://electronicbookreview.com/essay/the-anxiety-of-imitation-on-the-boringness-of-creative-turing-tests/.

[153] Porter, *Cambridge Introduction to Narrative*, 19.

The printed book itself is also a type of material and paratextual justification of the text that provides a sense of permanence both personally and institutionally due to the fact that it is easier to preserve than other types of digital media. The archival processes of digital works are more complicated and less standardized. If one does go through proper digital archival practices, work done in digital mediums have a relatively short shelf-life before they don't run on current hardware, or have to be ported to a new framework, or their hosting services stop working, or the files accidentally get deleted. Institutions also have an easier time purchasing and archiving printed works, as Johnston notes that his *ReRites* box set has been purchased by multiple rare book institutes that would not be interested in purchasing the USB of his collected multimedia poetry. Each of the author/programmers featured in this thesis has a history of focusing on multimedia literary art, but expanded their practice to incorporate print. *Articulations*, *1 the Road*, and *ReRites* are each the first literary project that these authors have published in the form of a printed book through traditional publishing with established presses. Allison Parrish had created self-published zines, but had not seriously considered writing for the medium of print. However, the affordances of the traditionally-published book quickly made themselves clear. The book, she says, is, "an object whose objectness is recognized by people," something that, "feels like an actual thing." All three author/programmers that are discussed in this study expressed a similar sentiment. Ross Goodwin of *1 the Road*, who consistently emphasizes print material in his work, insists that, "Seeing generated text in a physical format embodies it in a way that's important to making it legitimate." Through his experience presenting his pieces around the world, he's noticed how the knowledge that text is written by a machine already creates a sort of dissonance. Based on this fact alone, people will read phrases that are commonly used as if they had never heard them. Getting this kind of text to give people an emotional reaction, he says, “requires a frame for that content to exist in that the person can connect with it so that they can get over the fact that it was written by a machine.” Goodwin not only aims to provide a tangible frame for his readers, but also uses printed material to fight against, what he calls, “the sense that the digital doesn’t belong to us.” Similarly, David Jhave Johnston voiced his surprise at how satisfying the books as objects were to behold. He was “really happy to have this physical object,” and said it was “amazing how gratifying it was to just see something [and] say oh yea, it’s got [..] form. [Laughs] It’s got substance. It’s got cognitive validity. It’s strange. Like if you look at the size of a file online and it says it’s two gigabytes you think that’s amazing but [...] it’s not the same as holding a book.” Parrish confesses of the printed book, “I like it a lot. Maybe I like it more than I should. I would be totally happy just making books and static artifacts the rest of my career.” A print book never goes down, something that is a personal relief to Parrish, who used to work as a software engineer and would frequently be called in on weekends to fix problems with systems that needed to be up all the time.

The printed book also provides a sense of finality to each of these projects. “A book is a book and it stays,” Parrish says. “I like that it’s just finished, it’s done, it will always be done. I don’t have to pay to keep it online.” A print book never goes down, something that is a personal relief to Parrish, who used to work as a software engineer and would frequently be called in on weekends to fix problems with systems that needed to be up all the time. *Articulations*, *1 the Road*, and *ReRites* are all very offline and do not have ebook purchasing options. Lori Emerson suggests that the supposedly obsolete technology of the book is becoming a safe haven for digital literature, precisely "because its particulars cannot be tracked, monitored, indexed, fed into an algorithm, and given back to us as a commodity."[154] The "safe haven" of the printed book is especially relevant to a work like *1 the Road*, which so heavily emphasizes tools of surveillance, only to then bring the final work offline in a format that cannot be reabsorbed or "read" back into a network. The printed book in the context of this system is a final statement on the surrounding project that creatively subverts industrial tools of tracking and monitoring. In Goodwin's case, the printed book provides this additional meaning and is an optimal display for computer-generated text that conveys a sense of human connection and material belonging, recontextualizing the book in relation to everyday interactions with the networked and vectorized word. The reasonings behind the inclusion of the printed book into an electronic literature system are diverse and complex and deserve attending to.

For *Articulations*, *1 the Road*, and *ReRites*, the process of creating the work for a print format mandated at least some type of postprocessing or editorial compromises even if the artistic perspective of the author/programmer was specifically positioned against editorial intervention and postprocessing. This editorial aspect can be minimized, but should be expected and interrogated for readers or scholars of computer-generated books. The form of the print book also denies more contemporary modes of analysis along the lines of digital humanities and shapes the reading process. Due to the comparative nature of the work, it is tempting to want to search for word frequencies when reading *Articulations.* One could do this type of search of the database that Parrish thoroughly documented and made available online without any sort of vector representation or random walk through a vector space. Without being able to quickly

[154] Emerson, *Reading Writing Interfaces*, 184.

search, the reader is forced to take the walk of the algorithm and experience the repetitions, not through a count of word occurrences in the form of a single number, but through their articulated repetitions and contextual phonetic relationships. The printed book brings the focus of the reader to the text itself and a linear traversal through the text.

However, the printed book need not be traversed in a linear way. *Articulations*, *1 the Road*, and *ReRites* are all legible in the sense that you can sit down and read the text in the books out loud, which is not the case for all computer-generated literature. The narrative or reading experience might not be conventional, but each book consists of language that is familiar and legible in some ways. Many computational literature projects are intentionally illegible or so extensive that it is not realistic for someone to sit down with the book and read through the entire thing in a linear way. Some argue that computer-generated literature is meant to be sampled rather than read all the way through.[155] This mode of reading is also part of the history, not just of computer-generated literature, but also literary works that explore the potentiality of combinatorics without the assistance of a computer, such as the work of Oulipian writer, Raymond Quenueu, and his *Cent mille milliards de poèmes*, or *One Hundred Million Million Poems*.[156]

[155] Flores, *Distant Writing*, 2020.

[156] Raymond Queneau, *One Hundred Million Million Poems* (Kickshaws, 1983).

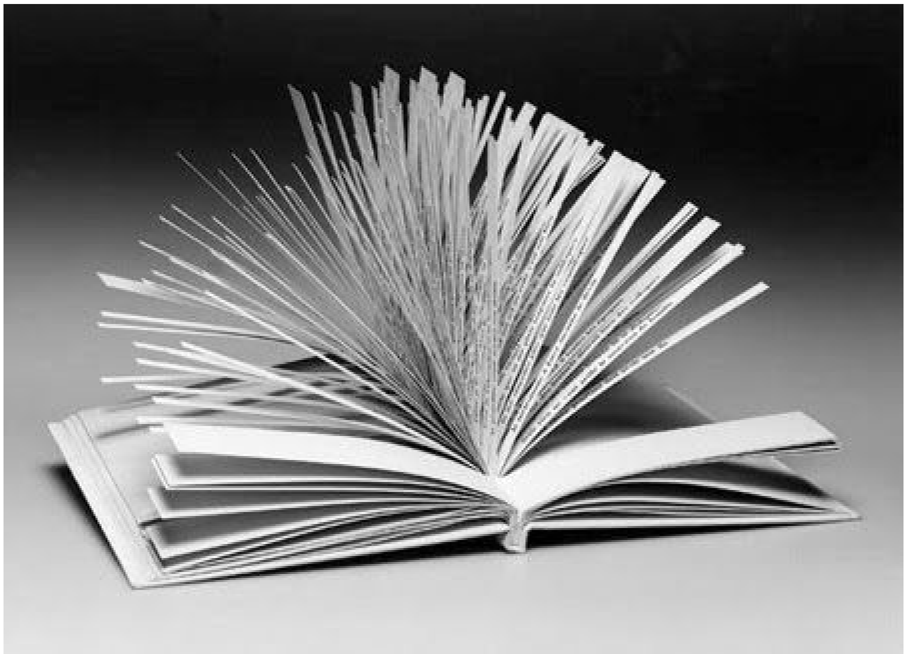


Figure 18. One Hundred Million Million Poems *by Raymond Quenueu. Each line is cut out so that the reader can manually permute lines and generate sonnets.*[157]

This text and its material format lend itself to a reading style that is one of sampling. Queneau wrote ten sonnets with the same rhyme scheme and sounds such that each line in each of the sonnets could be combined with any other line in order to create an analog text-generation system that could potentially create 100,000,000,000,000 sonnets. The book was printed with each individual line cut out so that the reader can operate the paper interface to create various combinations of lines and generate sonnets during their reading experience. This material manifestation of an algorithm also points to a unique relationship that AI-generated literature has to the printed book. Due to the high-dimensional spaces of machine learning algorithms, a similar material format in a 2-dimensional or 3-dimensional space would not be very effective.

[157] LEA, 2012. https://www.leoalmanac.org/vol17-no2-hundred-thousand-billion-fingers/.

Articulations, *1 the Road*, and *ReRites* each provide unique solutions to illustrating, traversing, or contextualizing some high-dimensional space in a linear format. This is one of the poetic challenges of AI-generated literature and results in different authorial decisions such as the random walk accumulation of *Articulations*, the time and location-constrained linear entries of *1 the Road*, and the embodied, year-long collaborative practice of *ReRites*.

5. CONCLUSION

After a close analysis of three books of AI-generated literature and their computational modes of generation, it is clear that AI-generated literature is not a monolithic practice that erases the author, but instead encompasses a diverse range of writerly practices. Rather than the genre of big data poetics minimizing the role of the author/programmer[158] or obsolescing the author from textual production,[159] machine learning provides writers with new poetic understandings that are explored at different levels of machine writing. AI-generated literature can also be read in conventional ways and is not impenetrable to interpretation. Through the documentation of writing and reading strategies related to AI-generated literature, we can begin to broaden this category of study and understand ways to analyze and interpret this work, despite the perceptions of the "black box" algorithms of AI. As the capabilities of AI and natural language generation technologies continue to change and improve, we can categorize the algorithms used to create AI-generated literature, compare approaches, and discover new ways of analyzing this type of work. More books of poetry and novels will be created using machine learning and labeled as AI-generated. I know of many such projects in the works, which, as they are being created, continue to change due to the fast pace of developments in new approaches, language models, and text generation tools that are available.

Initial reactions to AI-generated text often include accusations of it being too easy, or a type of cheating, or that it somehow destroys the concept of authorship. Rather than accusations, these are interesting questions that each of the authors in this thesis responds to in different ways. Parrish chose to use a dataset of text that is in the public domain in order to rightfully publish

[158] Rettberg, *Electronic Literature*, 53.
[159] Henrickson, *Towards a New Sociology of the Text*, 181.

and lay claim to the language in *Articulations*. She also considers herself a poet who makes the majority of her writerly decisions in the development of code and does not want to edit the output of the programs that she writes. Goodwin specifically avoided using source texts from the genre he was trying to emulate in response to certain ideas of cheating. He also adjusts the parameters of his models to ensure that his neural networks create truly novel output. Additionally, he runs particularly interesting parts of his output through a Google search to check for plagiarism.[160] Goodwin positions himself not as a poet, but as a writer of writers, emphasizing the work that goes into the development of the interfaces that he is interested in creating. Johnston, who considers himself a poet and digital media artist, but does not consider the poems in *ReRites* entirely his own,[161] developed an editorial process that focused primarily on the human labor that went into his machine learning lyricism.

The writerly interventions during the stages of preprocessing, coprocessing, and postprocessing that are collectively unique to AI-generated literature include the creation or selection of datasets, the development and training of a machine learning model, the decision of how to represent textual data in a vector format, the development of a text generation program that interacts with a trained model in some way, the selection and use of hardware and sensors, the inclusion of either one or multiple trained models, the control of and interaction with sensors, the parsing, formatting, and selection of textual output, the deletion of output, and the explicit editing of textual output. Author/programmers working with AI must make some decisions regarding each of these categories, and specific authorial practices emphasize creative explorations at different points in these stages. Parrish's work consistently emphasizes the

[160] Goodwin, Appendix B.
[161] Johnston, Appendix C.

creation of novel machine learning models. Goodwin has a sustained practice of linking machine learning to specific hardware interfaces. Johnston's work emphasizes the human as mediated by the machine, and his explorations in AI have focused on editing and postprocessing, something that Parrish and Goodwin both resist.

AI-generated literature is not a total break from traditional literary studies. We can use both established and developing modes of literary criticism to gain a deeper understanding of these texts. The material qualities of AI-generated literature apply not only to the printed book itself, but also to the text as it goes through a material trajectory of generation. For *Articulations*, for example, the material journey of each line of poetry begins during the poem's first publication and travels through the creation of the Project Gutenberg etext, the collection of the line into Parrish's dataset, the plotting of the line into a point in a phonetic vector space, the output of the line during the course of Parrish's selected pseudo-random walk algorithm, and ultimately the inscription onto a printed page. Additionally, paratextual elements of the book including introductions, explanations of method, response essays, and classification information contribute to the way we can read AI-generated texts. The close analysis of code can be used to inform an intentional reading of the text that focuses on what the author/programmer was trying to achieve, while a good old-fashioned textual analysis can also result in understandings of narrative, aesthetics, and underlying processes.

The choice to use AI, to engage with the idea of AI, is an editorial one as well. Such computationally complex and data-intensive programs are not always necessary to achieve a certain textual or aesthetic effect. There are rich histories and active communities of artists and writers working with text generation systems that specifically emphasize small-scale programs.

The three authors in this piece do not limit their work to just exploring machine learning, but also engage with other threads of practice emerging in computer-generated literature. Ross Goodwin has a piece published in Taper, an online magazine for small-scale computational literary art.[162] Both Ross Goodwin and Allison Parrish have submitted projects for NanoNaNoGenMo, a spin-off of National Novel Generation Month (NaNoGenMo) that prompts author/programmers to generate a novel of at least 50,000 words by writing a program that is less than 256 characters long.[163] While AI-generated literature and machine learning is a diverse and fascinating area of study in itself, the broader category of computer-generated literature is even more diverse. All of these categories and definitions also blur into each other, overlap, and cannot and should not be totally isolated. Still, it is useful to analyze techniques, practices, and modes of experimentation, in order to provide a vocabulary for writers and readers of these types of work so that we can communicate and collaborate across disciplines, professions, and interests.

INITIAL INSIGHTS INTO RELATED QUESTIONS

There are many questions that remain and emerge from the "opening" of AI-generated literature. There are also many tangential and related avenues of research to further open and contextualize the study of AI-generated literature that I did not include in this discussion. More work needs to be done in terms of excavating the history of artificial intelligence and its literary past. There are many future-oriented speculations about AI, but less holistic historical projects, especially as they relate to the intersection of AI and literature. Also of relevance to the literary history of AI is the portrayal of fictional textual automatons in literature. Representations of

[162] Taper, "Taper: an Online Literary Magazine for Small Computational Pieces," (Bad Quarto). https://taper.badquar.to/.

[163] Nick Montfort, "Nano-NaNoGenMo or #NNNGM," Post Position, November 16, 2019. https://nickm.com/post/2019/11/nano-nanogenmo-or-nnngm/.

computer and AI-generated writing have raised the curiosity of the literary consciousness for some time, including George Orwell's "Versificator" in *1984*, Roald Dahl's "Great Automatic Grammatizator," Fritz Leiber's "Wordmills," and Arthur C. Clark's "Word Loom" in *The Steam-Powered Word Processor* to name a few examples. An in-depth analysis of these fictional representations and the ideas surrounding them given cultural conversations surrounding computing at the time of their writing is another avenue for further research.

Another potential area of study is the proliferation of fakes in AI-generated text. AI has a long history of hoaxes, the most famous of which is probably the 18th century Mechanical Turk chess-playing machine that housed a secret human player, but includes many additional dupes, such as the French digesting duck automaton. This machine ate a piece of corn, digested it, and then excreted bird feces. It was marketed in the 18th century as a small "chemical laboratory" and an accurate model of mallard digestion, but in fact had a pre-loaded compartment of mashed bread crumbs that had been dyed green. Even the poop was fake.[164] This history extends to AI-generated literature as well. When a book is published that claims to have been written by a computer, but with no documentation of the systems behind them and results far superior than what can be achieved with state-of-the-art approaches, there is much speculation. Furthermore, manual or "fake" AI processes are often used for prototyping and can be an important component of the development of AI systems. Amazon's Mechanical Turk, a micropayment crowdsourcing platform that distributes immense amounts of tedious tasks to human workers over the internet, is often used for the purposes of AI research. Research in combating malicious efforts that are enabled by AI-generated text will also be important in the future, although will

[164] Annetta Black, "Canard Digérateur De Vaucanson (Vaucanson's Digesting Duck)." Atlas Obscura, January 30, 2010. https://www.atlasobscura.com/places/canard-digerateur-de-vaucanson-vaucansons-digesting-duck.

most likely be undertaken by computer scientists and those outside of literary studies. Still, if text-generation systems are used maliciously this will also be an important part of how we interact with and conceptualize these systems.

This history of frauds is also relevant to contemporary writing and the perception of AI-generated text. There has been an influx of people and brands claiming to have used AI to generate some text that is actually human-written.[165] While these experiments are usually written for some sort of comedic effect, they point to a contemporary understanding of AI poetics and a certain voice and tone that can be recognized as machine-generated. Electronic literature researcher Malthe Stavning Erslev has called this specific practice "bot-mimicry," and writes that these texts position the machine as something that is harmless and nonsensical while also something that is able to mystically extract some sort of crucial essence of a topic or a way of writing.[166] The "I Forced a Bot" meme that began in 2018 inspired hundreds of fraudulent computer-generated texts that were written humorously by people who were imitating a machine voice.[167] This was mainly in response to the viral project by Botnik Studios in which a bot trained on the *Harry Potter* books generated a new chapter for the series. This also resulted in brands co-opting the meme. Burger King released a few purportedly AI-generated commercials (actually human-authored) that were narrated by a synthesized voice.[168]

Text-generation algorithms also inspired the predictive text meme, in which users use

[165] Matt Schimkowitz, "I Forced a Bot," Know Your Meme, June 15, 2018. https://knowyourmeme.com/memes/i-forced-a-bot.

[166] Malthe Stavning Erslev, "I Forced a Bot to Read over 1,000 Papers from Open Access Journals and Then Asked It to Write a Paper of Its Own. Here Is the Result. Or, a Quasi-Materialist Approach to Bot-Mimicry," *A Peer-Reviewed Journal About* 8, no. 1 (2019): 114–26. https://doi.org/10.7146/aprja.v8i1.115419.

[167] Schimkowitz, "I Forced a Bot," 2018.

[168] Tristan Greene, "Burger King Trolls Artificial Intelligence with New Ads," The Next Web, September 28, 2018. https://thenextweb.com/artificial-intelligence/2018/09/28/burger-king-trolls-artificial-intelligence-with-new-ads/.

their device's predictive text tool to complete a prompt. Each user's predictive text algorithm for messaging is trained both on massive amounts of collective data and each user's own unique style of thinking and writing. Thousands of people online have used their custom algorithms to act as absurd prophets in a very pataphysical creative writing exercise powered by institutional corporate tools. The algorithms are being consulted as a way of understanding the self: "Type 'I was born' and then let your predictive text write your autobiography."[169] As a way of reaching into an unknowable future: "How about we use predictive text to write our own epitaphs?"[170] As a way of parodying already absurd voices: "Type 'You are not a real gamer until you,' and let your predictive text finish that quote for you."[171] These kinds of exercises are experiments in testing the limits of the tools at our hands, but also a way of understanding the machines and algorithms that constantly surveil us. These predictive text systems are practical and discretionary by design with a large amount of censorship built in resulting in "customizations [that] make us feel seen. The random quirks give our pattern-seeking brains delightful connections. The parts that don't make sense reassure us of human superiority."[172] Further investigation into popular conceptions and interactions with computer-generated text will be an important part of understanding text-generation and its influence on cultural perspectives and literary creation.

This thesis focused on the works of author/programmers who individually developed and

[169] Brad G., "Type 'I Was Born' and Then Let Your Predictive Write Your Autobiography," Twitter, November 3, 2017. https://twitter.com/therealbradg/status/926260526270652416.
[170] "How about We Use Predictive Text to Write Our Own Epitaphs? Write 'Here Lies [YOUR NAME]. [Preferred Pronoun] Was..." and Let Your Phone Finish the Rest by Just Pressing the Middle Text Button," reddit, January 2, 2018. https://www.reddit.com/r/funny/comments/7nlnd7/how_about_we_use_predictive_text_to_write_our_own/.
[171] Ludipe, "Type 'You Are Not a Real Gamer until You' and Let Your Predictive Text Finish That Quote for You," Twitter, April 4, 2019. https://twitter.com/Ludipe/status/1113717316536209408.
[172] Gretchen McCulloch, "Autocomplete Presents the Best Version of You," Wired, February 9, 2019. www.wired.com/story/autocomplete-presents-the-best-version-of-you/.

programmed the computational system that generated a literary output that was in some way authorially attributed to their own name. However, tools for writers such as the Botnik predictive text keyboard or talktotransformer.com open up a new area of inquiry, lower the barrier of entry to working with these kinds of systems, and will prompt a large number of AI-generated writing. Perhaps the writerly practices listed in this thesis will prompt ideas for new tools, or expand the ways that we can think about them if they do become very prominent in the creation of literature.

This thesis focused on English-language AI-generated literature, but there is work being done in many languages. One future direction is to expand the global reach of this type of analysis to incorporate and even begin translating works created in other languages. This also requires specific perspectives on what it means to translate a work of computer-generated literature and explorations about best practices to do so. More research can also be done regarding AI-generated literary work that is not in the form of a printed book. AI will be applied across the spectrum of genres in electronic literature, and we can compare the practices of AI-generated literary work in different mediums. A broader framework for cataloguing, categorizing, and thinking about AI-generated literature would be a larger project that would generate some important insights.

FINAL THOUGHTS

I started this project in the fall of 2018 when *Articulations* and *1 the Road* had just come out and LSTMs were the state of the art in natural language generation technologies. Over the past two years, more advancements have been made and AI-generated text is even more entrenched in cultural conversations. More works of AI-generated literature have been written, and are in the works, that will provide new ways of thinking about these technologies. I hope that this thesis will provide some useful context for the interpretation and categorization of these new works. I hope that this project is useful to literary scholars who are interested in computer-generated literature and to writers who are interested in using machine learning in their creative projects.

In the 2018 book *Pattern Discrimination*, Hito Steyerl wrote that the images generated by Google Deep Dream manage "to visualize the unconscious of prosumer networks," and "give those forces a face."[173] If AI-generated images give our systemically data-dominated culture of prosumption a face, then perhaps AI-generated text can give it a voice. Through creating a lineage of AI-generated literature and exploring the ways that each book's unique computational technique for writing comments on what it means to be a human in a network of bodies and machines, we can find ways to use these technologies in creative and compelling ways and to expand and reinvigorate literature rather than stifle and dehumanize it.

[173] Hito Steyerl, "A Sea of Data: Pattern Recognition and Corporate Animism (Forked Version)," in *Pattern Discrimination*, (University of Minnesota Press, 2018), 9.

APPENDICES

Appendices A–C include transcripts of hour-long conversations between me and the creative practitioners behind *Articulations*, *1 the Road*, and *ReRites*. I am grateful for the generosity of their time, their thoughtful responses, and their permission to include these transcripts in this work.

APPENDIX A

The following is a transcript of a conversation between Judy Heflin and Allison Parrish that was held on February 20, 2020 over Skype.

EXCERPTS FROM CONVERSATION WITH ALLISON PARRISH // FEBRUARY 20, 2020

Judy Heflin: Do you have the book in front of you by any chance?

Allison Parrish: I can bring up a PDF of the final proof.

JH: I was just going to start by asking if you had a specific favorite of the poems or sections within the prose poem.

AP: I do have the copy that I bring with me to readings that I have marked up with all the passages that I like to read from and then also the length of each passage when I read it, but I don't have it in front of me right now, so I don't know for sure. The passage that I have highlighted in my PDF starts with "Will she ever, ever, ever hither come. How every pleasure, every good," it starts on page 30 and it goes to the top of page 33, ending with, "The sun-like eyes whose light and life and like a child like a child." That's my go-to passage when I want to do an off the cuff reading of it.

JH: Is there any specific reason why you like it?

AP: There are lots of sections in the book where there is a lot of lexical repetition and I like that those are in there, but when I'm doing readings I want to try to avoid those, so this is a section that has a lot of phonetic repetition and has a lot of interesting language in it, but doesn't have a lot of actual repeating of words, and there are lots of little interesting transitions between repetitions of things, and then it has a little bit of an arc in it from a semantic perspective. I like ending on a phrase that feels like it's an ending when I'm doing a reading, and that, "The sun-like eyes whose light and life and like a child like a child," feels like an ending point.

JH: So you are open to people interpreting some sort of narrative or semantic qualities from this work?

AP: Did I come off as not seeming like I'm like that?

JH: Sometimes authors are trying to work against that and sometimes they're open to it. Are you interested in that sort of semantic quality?

AP: I think what I'm trying to do with all of the stuff that I work on is trying to create an artifact that affords those kinds of readings. I want to create something that is language that strongly suggests things, and among those [...] Obviously it's made out of words. If I didn't want it to have that kind of meaning then I would have made this art with some other medium. You read words and words have meaning, so regardless of whether or not I want people to do it, it's still going to have that suggestion and part of the intention here is to create interesting juxtapositions of words and phrases that call to mind interesting things. But I didn't plan any of that, there is no semantic structure to it in terms of the program that I wrote, but I am interested in this epiphenomenal meaning that arises from text that otherwise doesn't have any intentional semantic coherence.

JH: Can you describe your broader practice at the intersection of computation and writing? In general the sorts of questions you're interested in and what you find so interesting about using computational tools to explore text creatively?

AP: I see computation as the latest in a line of tools. The first step is to not think about computation as something new. Depending on how you define what a computer is, poets and creative writers have been using computational tools for hundreds of years to create interesting poetic artifacts. This goes back at least to Tristan Tzara in 'How to Make a Dadaist Poem' but then maybe back to middle age philosophers working with combinatorics. And maybe further than that, you can argue that written language itself is based on spreadsheets that Mesopotamians used. So there's this really ancient connection between language and computation and mathematics, and my goal is no different from the goals of the Dadaists or the Oulipo or even the Surrealists or Language Poets or whatever, it's like how do we create new forms of language? How do we get outside the conventions of language that are limiting the possible forms of expression and use the other tools that are available to us other than just purely the way that we feel like we should be speaking in order to create new ways of being expressive with language. So it's sort of, at its root it's kind of iconoclastic. I'm mostly interested in breaking things and computation just happens to be a really helpful hammer for doing that.

JH: Who are your biggest inspirations?

AP: Gertrude Stein is sort of the motivating [...] It was when my creative writing teacher in high school had us read ‘As a Wife Has a Cow A Love Story’ and that was sort of the reading that opened my mind up to language being something that can do things other than make the world keep going the way the world was already going to go. Probably the artist that I look at most is Jackson Mac Low. He wrote an essay called Poetry and Pleasure that is sort of an instruction manual for my practice, including how to go about it, but then also how to feel about it, how to position it politically. His work is the thing that I constantly look back to for inspiration and also as a guidepost to make sure that I’m doing interesting things that I want to do.

JH: Can you explain the concept of the vectorized word and what excites you about the vector representation of language?

AP: A vector representation of language is just like most of the work that happens in machine learning operates along statistical principles, basically in order to represent anything so that it’s digestible to machine learning you have to figure out some way to represent it as a number. The kind of number that’s most useful for doing that kind of computation is a vector, so just like a point in an n-dimensional space is the easiest way to think about it, like on a cartesian plane a vector is the x-y coordinate of a point, but then you have two, three, four, 50, 100, 1,000 dimensions and the math doesn’t really change. Most machine learning is based around the idea of like, for example, a categorization task is just like how do you draw a line between these points such that on the left of the line is “things belong to category A,” and on the right of the line is “things belong to category B,” and that math works the same, whether the things you’re representing are words or documents or photographs or lines in a spreadsheet for houses or cars or whatever. Vectorization is the operation that has to happen in order for data to be legible to machine learning processes. There are a couple of operations that vectorization makes possible that you wouldn’t be able to do otherwise, and that’s something like, the main one that’s interesting to me, is similarity or proximity. Once you figure out a way to vectorize words or vectorize sentences or vectorize whatever, it’s a very easy mathematical operation to say like show me all of the other things that are like this. And once you can do that you can do things like this word is represented by Vector A, this word is represented by Vector B, we can find the midpoint between those lines and find the words that is closest to that, so it allows you to kind of draw or infer gradients between words when they are turned into vectors. Those are new kinds of operations from a poetic standpoint that you wouldn’t be able to do unless you had some kind of mathematical representation of words, and I don’t believe that poets who have been working with math and combinatorics and statistics and stuff like that, I don’t think they’ve been doing those kinds of poetic operations before. So that’s really what was interesting to me was just that idea of like similarity, the idea of interpolation. These are new poetic operations that we haven’t had access to before.

JH: Is that why you chose the random walk algorithm for this kind of traversal of distance and spacing? Were you considering any other ways of exploring the vector space through different algorithms?

AP: Yes that's why I chose the random walk, because it was the easiest way to do that to make something coherent from this. I still am interested in this idea of showing things that are similar to each other in a row. Bringing out one thing after another that's just like slightly different from the previous one is kind of like the primary poetic operation of machine learning when you think about it. A lot of art having to do with machine learning is essentially like exploring a latent space or showing items that are similar to each other, and that has a lot of energy in it from a juxtaposition standpoint. When you show two things that are similar to each other right next to each other, then it kind of brings out both the similarities and the differences. And the random walk was the easiest way to do that to just show one item after another that was just a slight variation on the phonetic content from that perspective. It would have been possible to do something else with this space, like something that's exhaustive, I don't know what other algorithms I could have used, but like drawing a space-filling curve like a Hilbert Curve in the space would have been another way of going about it, or just like drawing a bunch of random lines that were connected to each other. It could have been like a series of interpolations between lines like I've done with other work, with the *Compasses* book that I published recently. So there were other operations that I could have done, but the random walk felt like the best way to produce 100 pages of text.

JH: Was that the goal? 100 pages?

AP: The book only came into existence because Nick Montfort asked me to do a book for his series. I hadn't done a book length work before that and it hadn't really occurred to me as something that would be desirable to do, but the book has to be a particular length, so the program that does the random walk I just set it to end after a particular number of lines, so it came out to about 70-80 pages in the final output for the phonetic similarity part and there is the syntactic similarity part that follows that's about another 25 pages. It was just an arbitrary limit because Nick didn't want these books to be much longer than 100 pages, just I think for printing cost reasons. I had the program generate I think like 20 different drafts and then I just picked the one that looked like it had a somewhat interesting beginning and somewhat interesting ending, and sort of did spot checks on them just to make sure that the formatting was more or less OK, and the one that's in the book is the one that I eventually picked.

JH: How did you end up choosing the forms of Parts 1 and 2, and even that there would be a Part 1 and Part 2? Did you think that the prose poem was a more interesting way to explore the phoneme features whereas the lineated verse was more interesting with parts of speech?

AP: Clearly the phonetic portion for me is like the main part of the work. That's the part that I still talk about and that I read from. I think it's more interesting in terms of both an aesthetic and technical perspective, but I thought that it needed a counterpoint, something that worked with the same means of determining similarity more or less, but in a different form using a different set of features. The thinking was that it would help you understand how the process worked better so that you would be able to look at Part 2 and then go back to Part 1 and say like "Oh I see how these two things work along similar principles." And then that would heighten your appreciation for both of them because in seeing the difference between the two you can also see the similarity in how they work.

JH: For the verse lengths in Part 2, are the stanza breaks programmed as well?

AP: To do that I developed a very quick and dirty statistical model of stanza lengths in Wordsworth. I just got like a big text file with Wordsworth poems in them and then did more or a less fitting a normal distribution to the number of lines per stanza, and then I used that to generate random groupings of stanzas according to those lengths. There's a lot of seven and seven, there are a lot of random groupings of even numbers of verse. I don't remember exactly how I did this. Some of them have evenly split stanzas and some of them are randomly chosen. The lengths are based on a quick and dirty statistical analysis of Wordsworth.

JH: Was there any sort of editing on your end after? Are these all just purely output of a computational process?

AP: That is a difficult and interesting question. The thing that actually happened is that I was very dedicated to the idea of not doing any editing after the fact. I wanted the only editing to be, as I mentioned, there were various outputs and I picked among those. After the final proof was in and they sent the books to be printed, I was reading through my proof, and I saw that there was an antisemitic slur in it, that came from a poem that Coleridge wrote to a friend. You just can't trust these 19th century Englishmen to not have terrible attitudes about things. So I talked to the publisher and we actually pulled the first run and I sent them a new manuscript that just had that line removed. I didn't rerun the algorithm or anything because I liked the rest of it. But I just didn't catch that line when I was reading the proof so we had to [...] the first printing of this was sent to the dump. I had to pay for the second printing, but I'm glad that I did because the artifact as it stands [...] there's still some stuff in there that I would not necessarily want to say out loud with my voice, but I feel like I'm OK being responsible for all of the language in this text, and that's what's important to me.

JH: Were there any other sorts of surprises in the work having a physical form as a book because you said before it wasn't something you considered. Is it something you consider now as more of an option or something to work towards? What's your relationship with computer-generated text and the physical book?

AP: I don't know. I like it a lot. Maybe I like it more than I should. I think the benefit of having a book is that it is an object whose objectness is recognized by people. A Twitter bot is also a thing, but people don't understand a Twitter bot as a thing that is separate from other things. Books also have their own way of being preserved, like, you buy a book, you put it on your shelf, or your library buys it and now it's on the shelf of the library, and now that is just a part of what is in the library. So unlike a lot of digital works where I feel like they have a shelf-life of five to ten years before they don't run on current hardware, or they need to be ported to a new version of the framework, or the files get deleted accidentally, or the service stops working that they're hosted to. A book is a book and it stays, and now I know that at least several hundred people own this book, and people can cite it, and I can get sales figures for it, so it feels like an actual thing, and I like that aspect of it a lot. I like that it's just finished, it's done, it will always be done. I don't have to pay to keep it online. But it's problematic because I work as a professor at the Interactive Telecommunications Program and this work is not on a screen, it doesn't react to your movements or anything like that. It's incredibly non-interactive, or it's only as interactive as a book is interactive. So that's kind of a drawback, that it doesn't advance that other part of my practice, but I would be totally happy just making books and static artifacts the rest of my career because they have these other affordances that I think are really beneficial and really easy. I used to be a software engineer and I would get called in every weekend, not every weekend, but you know, I would be on call to fix problems with systems that needed to be up all the time and I really hated that. And that's the nice thing about a book is that it doesn't go down, there are no midnight calls that say somebody's book isn't working.

JH: In your practice would you say that the concept of not editing the output is something that is important to you artistically?

AP: Yes I think it is. Because for me, what's interesting about working with computation for me is that the decisions that you're making are in the program. And I like the idea of poetic decisions being decisions that result from a list of instructions. And once you go in there and start editing, then it's more about a different set of poetic decisions that to me feel, are more like conventional poetics decisions that are about like, "does this sound right?" or, "Is this making a reference to something?" You're drawing on these other aesthetic decision making processes that aren’t as interesting to me. Those are also aesthetic decision making processes that I don’t have any background in making and I don’t have a lot of confidence in making. I don’t write conventional poetry. I think part of the strength of that opinion rests a lot on my lack of self-confidence about my ability to write a conventional poem. If I had more of that confidence I might be more into editing, doing the co-writing sort of thing after the fact. But for me right now, where I am with my poetic practice, I feel like I’m really good at making the algorithmic decisions so I tend to stick with those.

JH: Can you talk about the motivations behind creating both Gutenburg, Damnit and the Gutenberg poetry corpus. Why and how?

AP: That actually has a long, weird history. I was teaching a class at the creative writing department at Fordham University, and in one of the classes I wanted to talk about Google autocomplete poetry, like when people type in the first couple of words into Google and people make a poem out of the suggestions. So I wanted to be able to do that with my class, like have them do some poems that were based on autocomplete, but I didn’t want to have to have them work with Google. Because I feel like Google’s autocomplete is a user-interface thing, but it’s also normative, when you type in your search request it’s not just showing you what other people have typed, it’s also showing you what you should type. And that feels sketchy to me from an educational standpoint but also from a poetic standpoint because when you publish a poem based on that you’re perpetuating Google’s values. So I wanted something more neutral, or at least something that had a different set of values. It occurred to me that since we were doing poetry, I could make a poetry autocomplete interface, so that’s what I did. I went through Project Gutenberg. I found all of the poetry and I made a simple program that does autocomplete based on lines of poetry in Project Gutenberg. So it was really that one exercise that led me to write the code that extracted that from Project Gutenberg. Once I had that dataset I used it as the basis for a lot of other experiments including articulations, and it worked really well for this. Those projects like Gutenberg Damnit and the Gutenberg Poetry Archive are me taking this a step further to where it’s a thing that has been released and I can do a presentation against and people can link to it and cite it. That was just like how do I maximize the academic prestige that I can get out of this thing that I did. That ends up being about 90% of my work is just like, one weekend in four hours I did this thing that is really cool, now I have to turn it into a paper or presentation or something, how do I do that. So that’s why those things have names, that’s why they have GitHub repositories. I wanted to go the extra step to make sure that it was shareable, citable, and so that other people could use it as well. And other people have used the Gutenberg Poetry archive at least for other projects, which is fantastic.

JH: So Gutenberg Damnit was a way of getting to the Gutenberg Poetry corpus because that was what you had in mind originally?

AP: Yeah, that's sort of the thing that I had to write in order to make the Gutenberg Poetry corpus, was Gutenberg, Damnit.

JH: Is there a reason you focused on the metadata category of poetry as opposed to fiction, or data?

AP: I'm really interested in the line as a unit. The line of poetry being analogous to a text file line being analogous to a database record. The line has this weird history that goes from Mesopotamian business records through medieval manuscripts where they needed to be able to count how much a scribe had written, up through Western poetry that uses the line as a unit, through to text files in the modern computer era where the line is also a unit both of semantics and of layout. I think that's a really interesting history, so the poetry corpus was just like what I wanted is just like a big thing that I could randomly pull lines of poetry from. But I've used the Gutenberg canon for other projects where I've been doing narrative things. It's really easy to go through and grab things that are novels, so you can recombine sentences in novels and stuff like that, so it's not necessarily limited to poetry but it just made sense for this. The other thing that's interesting about the Gutenberg poetry corpus is that it has that conventional diction, the diction we associate with capital-P poetry like high school class poetry. It's a double-edged sword because on the one hand you're exhuming the voices of these often extremely terrible men saying extremely terrible things, but on the other hand, when you make something computational with this corpus, you instantly have that, everybody recognizes it as poetry. You don't have to do the work of convincing people that it is poetry because it already has the diction that suggests that and that's a really helpful shorthand when you're doing computational work.

JH: Why phonemes? How did *Articulations* begin as an idea?

AP: There was a conference in 2015 or 2016 called AltAI where Mario Klingemann gave a talk, and I also gave a talk, I was in the audience for it. I gave a talk at the conference and then I also saw Mario's talk, and he was using vector representations of images, just similarity lookups, to build DaDa-esque contraptions out of clip Art as I remember it. And this project is so old and Mario is now like an art world super-star, so I don't even know if this project is on his website anymore. Just seeing those operations of similarity was really inspirational to me and I remember writing down in my notebook, "What is the poetry equivalent to this?" Based on that I got to work and tried to figure out different ways of representing words as vectors. I'd been doing a lot of experiments with pre-trained Word2Vec or GloVe vectors for semantics, but I was thinking that this would be more interesting if you could do it with sound as well. Even now, there's less research on the phonetics of words when it comes to machine learning applications. I was just playing around trying to do these random walks and interpolations between items in a vector space and I was thinking well how do I represent the sound of a line of poetry as a vector, and then that's when the idea of representing it as counts of phonemes came up and that sort of worked, and then it was like counts of bi-grams of phonemes, and that also sort of worked, but the key thing that came up that actually made it work the way I wanted it to work was to actually break it down into the phoneme features instead of whole phonemes, having bilabial and stop and all of those present in it. That was the thing that made it possible to get the effects that I wanted to get. So it really was just listening to Mario's talk, knowing about word vectors, and then figuring out like, I had the mathematical knowledge and tools to do this in this different domain, but what could I do with it there?

JH: The book is categorized on the back as POETRY / ARTIFICIAL INTELLIGENCE, and I was wondering is AI more broadly something you engage with as a practice? How do you think the label of AI hurts or helps this kind of work? Is it interesting artistically?

AP: It's helpful when I'm trying to get money, because there's a lot of people giving money to people doing artificial intelligence artwork right now. So, you know, I will use *Articulations* as part of my application when I'm applying for a grant or something. And the work that went into *Articulations* has spun out into different projects. I'm working with Google right now on a newer version of this model that can work with made up words and produce neologisms and stuff like that. I'm interested in machine learning because it is a powerful tool for producing the kinds of juxtapositions that I'm interested in. Artificial intelligence is such a hard phrase because it really is completely bound up in labor, like people's ideas about the future of work. The first place that people's minds go when I say that I'm working with poetry and artificial intelligence is they think like, "Oh, you're making a poet bot that will put traditional poets out of business." Because that's what AI is designed to do, it's designed to give C-level employees or companies an excuse to fire knowledge workers. That is the primary rhetorical and political and economic use of artificial intelligence, and I'm not interested in legitimating that particular line of thought. The work that I'm doing, I feel like I'm just doing the work of a poet. I'm using techniques to write poetry. I'm not trying to subvert poetry as a genre. I'm not interested in putting poets out of a job. I'm interested in being a poet and creating poetry. So calling it artificial intelligence is just a rhetorical move to be provocative in places where it counts to be provocative, but I'm interested in vector representations and machine learning ways of doing prediction, I'm not interested in artificial intelligence as a thing. I would never consider attributing the work to the algorithm. On the cover it doesn't say Allison Parrish and PoetryBot3000, it's just me, it's my work, I wrote it, I wrote the programs to write it. I'm not interested in machine intention or volition or communicating with machines or any of those other tropes that go along with artificial intelligence.

JH: Do you happen to have the code of the random walk algorithm?

AP: I can try to find it. Do you actually want to see the code? Do you want to know how it works?

JH: Not how it works, but how it was implemented. What happens at each step and how did you deal with adding randomness?

AP: Calling it a random walk is maybe not even technically correct, because a random walk is usually like Brownian motion. What it's actually doing is it picks a random vector from the vector space, and then, keeping track of things that it has already seen, it picks the closest line in the vector space and then does that in a loop; picks the one that's closest to the one it just saw, and then keeps on going up. The thing that happened with that is that it would sort of get stuck in a nearby cluster and then never escape from that cluster until it exhausted everything in that cluster, and then it would sort of like hop to the next cluster without showing anything in between. So, additionally, there's a velocity vector in there that's just a small, a uniform random number, that at each point gets added to the vector when it's picking the next closest just so that it's heading in a little bit of a different direction and that was enough to pop it out of that cluster. So that's the way I got around that. And also, like I said, I cherry-picked the one that seemed like it had the least amount of [repetition]. I wish that I had the know-how to make a better-reasoned, a more statistically motivated solution to that but that's the way that I picked it.

APPENDIX B

The following is a minorly-edited transcript of a Skype conversation between Judy Heflin and Ross Goodwin held on February 25, 2019.

EXCERPTS FROM CONVERSATION WITH ROSS GOODWIN // FEBRUARY 25, 2019

Judy Heflin: Similar to AlphaGo coming up with a very creative move that people wouldn't necessarily think of, a lot of what is exciting about what you called "narrated reality" is the instances of novel narrative or linguistic turns. You talk about this moment that you like in Sunspring where the camera angles change in a unique way. Are there any moments like that for you in 1 the Road? Which parts do you like the most?

Ross Goodwin: Totally there are. I'm just going to pull the book out because I think I have a few sections flagged. The end for me is really like one of my favorite parts for sure, like the very last paragraph. I just think that the way it ends is super interesting. Just in general, the references to kind of mundane scenery that are then riffed on in interesting ways can create some really interesting moments in the narrative. So I really like [...] the Great Pee Dee River bridge section is one of my favorites because of the repetition. I mean, you think about certain things when you're prototyping something in your apartment versus when it's actually in motion on a road. When I was prototyping I was like oh it's gonna be moving and it's gonna be different every time and you don't anticipate getting stuck in traffic and how that can affect the story. There's a part on the second or third day that begins with the Coca-Cola factory. My favorite parts are the parts that conjure imagery in ways that are relatable in a realistic sense, and those especially where the imagery contrasts in interesting ways with the data that the system was absorbing. The other thing about the book in general is I think it really is, the text in certain places feels very oppressive in interesting ways. There's the mention of concentration camps and there's a police car here and there, and the imagery is sometimes that of despair.

JH: Yes there are a lot of people standing around mysteriously.

RG: Yea exactly, and I think it's representative of a machine that can sort of place an object into a scene in an interesting way, but then doesn't really know what to do with it. It's pointing to this notion that the machine on its own has trouble moving between actions or moments in a story the way that we would consider them.

JH: You've described this as a rapid prototyping project. If you were to make another version of this would you include something to try to make the machine build a narrative once it puts an object in a scene or do you like the machine's inability to do that?

RG: I think that what's interesting about this project in particular, one of the things is the way that the journey itself sort of adds that in a way. It doesn't completely add it that in the same way that a person intervening would, but it was sort of my intent to play with that limitation. I am working on a new version of it that hopefully I'm going to drive this year. That's sort of in progress right now with some support from Google again. This time I want to do it with a bus, so the working title of the last one was WordCar and the working title of this one would be Wordbus. I've been joking that the name of the book would be *2 the Bus*, but I don't know if that'll come to fruition. I mean, I think the project is going to happen eventually, but I don't know what the name is going to be. One design choice I made when I was building the first version of this that I executed was I didn't want to generate an overwhelmingly large amount of material. I wanted it to be fairly sequential and sort of more sparse. I wanted to get something that wasn't so long that it was going to be completely unreasonable. I didn't want to give any publisher or anybody who got the manuscript the opportunity to heavily curate down to like a really really high-quality book. Because you can do that with this type of text, if you generated something 100 times as long as the book that I generated and then curated down to something the length of the novel that came out then you can actually get something a lot better than what came out on its own. What I wanted it to be representative of, at least for this first one, was like really kind of the raw output rather than any sort of curation. It was about pressing a button and making it happen rather than building in all of this extra work that would happen later. After doing it I sort of regretted that. I think the result is still great, I'm really happy with it, but for the next one I want to have multiple systems in the car going in real time at the same time simultaneously, so the next time I'm just going to have it continuously running, rather than going every 30 seconds like it was in the car in 2017. And we're gonna do a longer trip, so I think that will result in a much longer manuscript. And then it'll be a matter of some sort of automated curation technique to get it down to the book length version.

JH: What sort of curation did go into *1 the Road* if at all?

RG: The publishers formatted it. So far as I know there's no editing. They may have shortened it in a few places which I was sort of expecting. As a practitioner doing this sort of work it's very hard to find a collaborator whether it's a publisher or a filmmaker or whoever, who is not going to be like "Oh this paragraph doesn't work we can't do this delete." So there are always those compromises. They removed some but they didn't remove much and any editing they made was purely subtractive, but the other issue is that because I took such a rapid prototyping approach for this one I found a bug in my code after the fact that could, I'm not positive it did, but could have resulted in certain passages only appearing on the printed roll of paper and not in my computer's hard drive, so basically it was an exercise in both conscious lack of curation given that it had to be a publishable book and also consideration of that aspect as well.

JH: A lot of your systems are very focused on printed material, so WordCamera, Benjamin, and this as well all have specific printing mechanisms:

RG: They're all the same printing mechanisms actually. I use the same ones I have a couple of them.

JH: Why are you drawn to this specific printer and what you think is lost or gained when you end up with a printed linear text.

RG: OK, so I have some complex thoughts on this one. I used the same printer for all of them because after a lot of searching and buying random shit on ebay I discovered a printer that I really liked and that I could take the case off of that came with a fairly sturdy case that I could get for around $50 and that is wide format. Because the problem is you don't want to give somebody a manuscript and for them to associate it with a sales receipt, which are usually narrower. Aesthetically I think it works better if it's a little wider and the text can be a little bigger that way. So I use this particular printer because it's wide format and I can get them cheap and they're pretty ubiquitous. I also find it fun that it's the same kind of printer that cops use for traffic tickets so it's sort of a subversive inversion of a piece of institutional technology that's like really mostly used for that purpose and for like people in shipping warehouses. It's sort of like a capitalist tool in many ways. In terms of why I use them it's really because...I mean I started using them sort of on a whim just as like an experimentation when I was in graduate school and it was specifically with WordCamera and I figured that you know, a text polaroid, so to speak, would be an interesting paradigm to represent the project. The reason why I continued using them is after doing the prototyping for that project I found that seeing generated text in a physical format is very much [...] it embodies it in a way that's important to making it legitimate. If I give you an arbitrary line of generated text, to like an arbitrary audience member or a person who's reading it, you need to give that person a way to connect with it if it's not perfect, and even if it is really good then the presupposition that it was written by a machine is going to create a sort of dissonance regardless. A really good example of that is the line in Sunspring, which is "I was going to be a moment." So like, when *Sunspring* came out [...] Elizabeth Grey says that in her monologue [...] a lot of people were like oh well that line's so interesting, it's so interesting that she's going to physically be a moment, but no, people say that all the time, I was going to be a moment. It's just that because it was computer-generated and the way she intonated it, which I think was a really cool decision that she made, it sounded more dissonant and that expectation combined with the performance creates this sense that in fact doesn't make sense the way you think it should. So like, generative text has those qualities sometimes and making it give people like an emotional or the kind of reaction that someone would get from a traditional line of poetry I think requires a frame for that content to exist in that the person can connect with so that they can get over the fact that it was written by a machine. That's the main reason, and I've just seen a lot of evidence of that exhibiting this work. I haven't done any scientific studies but like it's the same mechanism. WordCamera sort of does it twice because it's both a photo you took and a physical receipt that you get to keep, so like both of those elements are important, the physicality of the text is also important. I would actually compare it to like reading computer-generated text aloud, which can sort of have the same effect in many ways because you're embodying it, you're giving it substance and in many ways you're giving it meaning.

JH: If it's just text without context on a screen then it's less embodied? Does something about the screen make it less official?

RG: Yeah because, so we in 2019 are all familiar with the digital and it's not really a foreign thing to us anymore, and digital objects, we can consider them alongside physical ones but fundamentally I mean they're in different categories I think still. And it obviously depends on the age group you're dealing with, I think kids have an easier time with embodiment of digital objects in my experience but for people my age and older I think that data on a computer is not the same thing as an object that you hold in your hand that you can look at and look back to and it's the same as it was before. We also have this sense that the digital doesn't belong to us I think, or a lot of us do.

JH: What do you mean by that?

RG: I mean like you know companies like Facebook, and the company that employs me for the mostpart Google, and like basically you know unfortunately AdTech by and large and the way that consumer data is collected I think has created this culture wherein we assume, or some of us, or many of us maybe, assume that if my information is on my computer, if someone takes my computer like I've lost everything if it's not backed up, and if it's backed up well then it's sort of like there's all of a sudden multiple copies of it that I'm dealing with, you know, which one is the one, and if I've backed it up with somebody, who else can see it now? In theory, I mean obviously I'm not paranoid or thinking that people are looking at my email, but I'm more just postulating that you've broken down this barrier where you are sort of like granting that information to a third party in exchange for an assurance that it will be stored for you. In a way it's like putting a receipt in a safety deposit box at a bank rather than on a magnet on your fridge, but it's not quite that because of the duplicative abilities of digital objects. It's a very interesting and complicated question I think.

JH: A lot of AI text generation systems are intentionally given very human names or acronyms that drive people to treat them as human, and you've done this in the past with your Benjamin system. If you look in articles a lot of writers will treat the output of Benjamin very differently from the way they treat the output of WordCar. I was wondering if you could speak a little bit about naming your systems.

RG: Just to predicate my answer to this, I'll just say that I'm really against human-shaped AI, except in the context of artificial limbs and very specific medical systems. I think that AI really shouldn't be human-shaped because I think it creates a lot of false expectations but also a conflation between the interactions that we have with machines and the interactions that we have with people. Fundamentally, when you're interacting with a person, that person has a different set of needs and affordances so-to-speak than a machine. I think that people are quite nasty to their digital systems even now, and if you create a human-shaped version of that you're creating this expectation and training certain people to be nasty to human-shaped things, which is not great. In addition to that I think there's a lot of [...] it's sort of the pink elephant or the white elephant in the room so-to-speak with respect to human-shaped AI is that most people who are making human-shaped AI or who are striving toward that, or a lot of people at least I think are doing it so that they can have a fantasy or that they can fulfill some kind of control fantasy, sexual fantasy. There's no design reason to make an AI human-shaped. People will argue like well, the world was built by humans, so something that's human-shaped has an easier time getting around. But that said, consider like insects, you know, they have a much easier time getting around than us. I just think in general it's really problematic, especially when you look at examples like Sophiabot, they're all women, and most of the people who build them are men. The other problem is that the word "robot" literally means slave, so then if you embody that as a person you're basically saying that in a way it's OK for something that's like a person to be a slave.

Naming Benjamin Benjamin was certainly a learning experience in like, "Oh, if you name it a human name, people will forget that somebody actually made this thing and it's not just a sentient piece of software that made itself. It's a problem and creates a false perception of AI in real life that furthers its conflation with science fiction. In science fiction, everyone has the robot butler that serves them drinks and that AI is always human-shaped no matter what, even if it's AI that is doing something that just dramatically does not require a human shape, it just is like inevitably a walking-talking person with arms and legs. I think that we're very programmed as humans to anthropomorphize and we're very anthropocentric in general in that we draw faces on rocks and think they're cute. There's lots of reasons why we project a human aura onto something, but we can't, or I think it's problematic when we forget that it's a projection and not in fact an actual person. And this goes back to the Dijkstra quote of, "asking whether machines can think is like asking whether submarines can swim." To ask whether machines can "blank" when "blank" is a human term is like asking whether a submarine can swim, fundamentally.

JH: How did associations with Google affect the work, if at all?

RG: It really hasn't significantly, like obviously I don't only make art projects that are personal expressions of myself. I work on other artists' projects sometimes and I make stuff that is not purely art for clients, and that's always been the case even before my relationship with Google began. What's really interesting about Google is that they seem really committed to pioneering a new model for supporting artists who work with tech, which is really great I think. You can't sell tech art in a gallery, generally, the same way you could sell like a painting or a photo, or a sculpture. It's a lot trickier to use that model to sustain a practice in tech art just for a number of reasons. I think that it's important if we want tech art to exist at all, for the tech companies that really prop up tech in general to support that, because it really is about breaking the mold on how engineers think. I think that engineers are taught in a very narrow way sometimes and I think that for us to not be stuck in that ontology and in that mold, and for people at large tech companies to not be stuck in that mold, is good. They need to be able to think beyond the scope of just how you solve a problem as an engineer and consider that some things aren't best approached that way and when you stop approaching everything that way you make a lot of insane discoveries in the same field that you were working in before sometimes. I think that engineers need to be more playful in general, or they could stand to be anyway, and more willing to think outside preconceived notions of best practices on how to operate.

JH: I feel like a lot of the sensationalism around AI is designed specifically to drive marketing and hype and your book is marketed as the "first real book written by an AI." What kind of conversations did you have about that and were you involved at all? The definition of AI changes so frequently that maybe it is the case that this book is the first one written by AI by today's definitions. I was wondering what your perspective on that is.

RG: I think you've already figured it out. AI is such a problematic term in terms of defining it for a given time period that it is true and it's also not true. I have publically disputed that it's the first book written by AI. If you see the *Atlantic* article that came out about the book I was pretty clear in response to the journalist on what I thought about that. I'll say what I said to him which was just like, my publisher insisted that it was defensible and they thought it would help the book sell, I convinced them to put it on the temporary sleeve rather than on the cover. Essentially it points to the easiest way to market something that was created by AI that exists in a genre that hasn't really been explored as much, but I think that the reason why I dispute it and why I would never claim that it's the first is because I think it just fundamentally downplays the work that my work is built on. A lot of times I feel like I'm standing on the shoulders of giants in many ways, so for me to be like "Oh no I'm the first," feels a little presumptuous.

JH: Who are the giants to you?

RG: Oh you know, I mean, in terms of computational creative writing, some of them, when I say giants I don't mean these are [...] because I would include William S. Burroughs in that category, and he's like kind of like also, you know he like murdered his wife, he's not a giant like in a good way necessarily, but in terms of this particular field, like the cut-up method, every member of OuLipo past and present. It's funny because I know actually a member of OuLipo Daniel Levenbecher, and I was talking to him recently and he was telling me about how technophobic the group is now, which is really funny to me. But anyway, who else, it's really also about the history of computing, because a lot of history is tied together if you talk about AI. It's not just Ada Lovelace and Alan Turing and the history of computing, and when I pin down Burroughs I'm really talking about the thread that's computational creative writing and not AI, which I would distinguish even though they sort of become the same thing at this point, in some instances at least. In terms of the history of AI, I think it goes back even further than the history of computing. In my opinion it sort of begins with medieval automata, and like, you know the pooping duck automaton from the 18th century, that French pooping duck? There's this really famous, it's been lost you know it burned up or something hundreds of years ago but if you Google it it's like pooping [sound of mechanical keyboard typing] pooping duck [...] uh digesting duck is the Wikipedia page article and it's created by Jacques de Vaucanson in 1739. What it was was this duck with mechanical innards and it would eat little pellets and then poop, and it was actually a hoax, like it didn't actually poop out the stuff that it took in, it had an additional hidden compartment, but it was still very mechanically impressive for the time. And the reason why people wanted to see it and why it was built as an attraction and people cared about it at all when the guy built it, is because at the time, and you can follow this thread through to the present day, people thought that if you built a mechanical duck, you could understand better how a real duck might work, and the problem is that like that thread—there's a great book about this called *Edison's Eve*—but like this notion that we can understand what a human is by building a mechanical person is just, it's silly. It might be true if we had the same technology or the same like even a vague understanding of how systems like the brain work. You know, the neural nets we have are really cartoons of brains and more analogous to insect brains, and you know, we can try to understand how the brain works by building models of brains with computers, but I think we also have to consider that like any conclusion we come to as a result, once that system becomes far enough removed from an actual brain, it might lead us to false conclusions pretty easily. So that whole thread of like trying to understand nature by building models of it [...] And like biomimetics and like making models of nature is a totally legitimate field, you just have to consider the bigger picture and figure out if you're really going to understand this entire system better by building a mechanical version of it, because the system itself might be a lot more complex than whatever simulacrum you're making out of it.

JH: The history of AI systems includes a lot of hoaxes. You said in an interview that you didn't want to train on Kerouac or other American road-trip material directly, because you felt like it might be cheating in a way. Across the board I think a lot of reactionary criticism to AI-generated text revolves around this idea of cheating. How does this perception affect your work, how do you work around it?

RG: I'm glad that I didn't include Kerouac in retrospect. At the time I was choosing the training corpus I just thought that it would be too on the nose, but after reading the output I was like "Wow this kind of sounds like Kerouac," because it's like a road trip thing and it didn't include anything from Kerouac in the training material which was sort of remarkable. So I really really loved that aspect of it, so I was sort of noting that retrospectively in the piece. In retrospect when I think about it it would be cheating to use Kerouac. If you want to make something that people were to compare to Kerouac, if that's the intent, which it wasn't even necessarily my intent to be honest, I wanted to make a road trip novel, which wasn't just Kerouac it was like a broader category.

JH: But you didn't include work from any of the other road trip authors who inspired this right?

RG: No, there weren't any of the other authors that I had in mind, I was just wondering if like, if you train on like arbitrary literature and you frame it within this road trip context, do you end up with something that's at all similar to anything created in that genre, and the answer is, you know like, maybe? Which is a really funny answer.

JH: Do you think that the accusation of cheating is a viable criticism?

RG: No, I think it is, I think that you have to be transparent with systems like this, because otherwise you are just sort of a glorified magician. Sometimes I feel like a magician anyway, but that's only because of the reaction I'm getting out of it. I always try to be very up front about what's happening. The thing is that like it is the case that people are used to [...] That's a really interesting question by the way, just like I haven't thought about this in the context of the entire history of hoaxes in general. I guess like you know, sometimes I feel like I'm peddling something, like you know like the tonic dealers of old, like a hundred years ago, who had their medicinal tonics that cures everything, and sometimes when I get near product people who want me to do things that are more commercial, and when I consider doing things like that I start to feel more like a tonic peddler because in a way, it's not that what I'm selling is fake or not legitimate, it's more like any marketing around this, because of the way that AI is perceived in the present as sort of like a panacea, is going to make it seem like "Oh it's a miracle, look at this text come out, oh my god it will solve all your problems, it'll solve your writer's block!" It's very sensational, it lends itself to sensationalism, like the idea that like something that isn't human is capable of writing even one coherent sentence.

JH: I want to know more about how you created your training corpus. Was it specifically arbitrary like you mentioned before or did you intentionally include any of your own original writing, was it all literature, and do you have any specific approaches for putting together datasets?

RG: Yeah, so the first question, it definitely wasn't super arbitrary. I, on the road trip like um, I didn't train any models that were specifically for the road trip, I have this pretty deep library of models that I'm constantly adding to, and I was switching between three primarily on the road trip. I was seeing the text come out in real time so I was like Oh, this is getting a little dark, let me switch to like this model, and I didn't do it that often. There were three distinct models that wrote the material in the book. One of them was like mostly literature, actually entirely literature, actually very Eastern European literature, it's actually like I'll just tell you, it's like, I created this one for this painter in Berlin named Roman Lipsky and I trained it on all of his favorite books, and so it was very kind of like bleak and Eastern European literature. The other one that's kind of used for less of it is like science fiction prose and poetry, and then the third one that's used for even less is like poetry from all of the world. So those are the three models that I like sort of instrumentalized in my practice usually. The thing about any given language model is that if it's trained on a large enough corpus it can really represent a lot of different kinds of writing. So I have, for like literary applications and poetic applications I have three models that I trust and I go to a lot for projects like this. And because it was a prototyping experiment, I wanted to use models that I sort of vaguely knew what they were going to say or the kinds of things they usually said. So basically yeah, that's sort of the composition of the training corpus. It was, I didn't take a super active role in like switching them constantly or anything, it's not like every other paragraph by any means, it's more like, there's like a few moments in every day in the book where you're hearing from a different model. The thing is that they're all being seeded with the same input structure, so you don't really notice that it's a different voice really because the voices are so determined by the way they're seeded.

JH: When you change between different models, is it mostly just based on tone?

RG: It was based on [...] when I say switched between models I don't see that as the same thing as curation because it's like you're not actually controlling what the machine says you're just controlling the kind of words it's going to use. You're right though yeah it was mostly about tone, where I was like, there's one model that wrote most of the book and what I wanted most of the book to sound like was pretty bleak, which I think is pretty appropriate for the time-period the book was created in and the content itself and the route, it's kind of a bleak route. The history of it even is very bleak. But the thing was that there were moments when I was like this is getting too repetitive I want to mix it up a little bit, but I was definitely eyeballing it, because I didn't have the whole book laid out in front of me on my computer, I was just sort of passively reading the text snippets as they came out and I was like “eh you know this seems a little like too monotonous I’m gonna like mix it up a bit and like switch models.” I do that at a couple of times at various points but like I don’t even remember, I could probably trace back where I did it, but like not super easily to be honest.

JH: Were there any parts of the book that you could recognize a specific part of your corpus or a bubble up of a recognizable text or author?

RG: There aren’t any things that I recognized. I would say the “street” is one of the most common words for an LSTM to write because “s” is the most common word beginning, “t” is the most common next letter, so like “street” is a very easy word for an LSTM to write in a lot of conditions. But, I find in general, like I try to generate the text at like temperatures, meaning like riskiness levels for the predictions, that result in predominantly original text, you know. I’ve seen instances where little bits of the corpus get parroted back but like, if you lower the temperature enough it happens a lot, but I try to prevent that at all costs. I think originality is important.

JH: So how do you choose the right temperature?

RG: A lot of times I outright screen the output to make sure the lines aren’t the same as any lines in the input. But like you said, there’s not like a really reliable [...] you can basically like keep drilling down and like yeah for any given word pair in *1 the Road* I really couldn’t tell you the likelihood that it exists in the corpus but any given trigram or four-word group, it’s like I’ve found really stunning examples, and I always Google them when I do, when I find something that’s really stunning, to make sure that it’s not something else. I always have paranoia that it might be. But yea, no, very very rarely have I seen an LSTM like really plagiarize and usually when that happens it’s because you’re at too low of a generating temperature.

JH: Why specifically are you drawn to LSTMs and also, the torch-rnn system?

RG: Because it’s a killer implementation, it’s a really good starting point because like [...] it’s a little outdated now I’m actually trying to find an alternative at this point because there are some other new implementations and architectures that are interesting that I’ve been trying out more lately. But torch-rnn in 2016 when it came out like [...] because prior to that, the best open source implementation in torch at least, and TensorFlow didn’t exist at this point and neither did PyTorch, was char-rnn by Andrej Karpathy and that came out like early 2015, and that was really good, but his lab partner Justin Johnson made torch-rnn and torch-rnn was seven times more memory efficient than char-rnn, so it’s still I think the most memory-efficient LSTM implementation on GitHub by far. So basically, with torch-rnn, you can train a much larger model on a much larger corpus than you’d be able to train with just about any other implementation that's publically available on a single GPU. So basically if you have one GPU and you want to train like a decent-sized model, torch-rnn gives you the ability to do that, pretty much right out of the box, so it's a great starting point for people who want to experiment with the technology.

JH: With your Google funding do you get access to things like GPUs or datasets that wouldn't otherwise be available to you?

RG: Yes. I'm at Google two-thirds of the time now, so I'm a contractor still but my involvement with them has increased, so I do have access. I have a Google computer and I have access to a code base and other stuff as a collaborator. I use torch-rnn for personal stuff still and stuff that's not at Google, but at Google everything I do is pretty much in TensorFlow. So, torch-rnn is still a great open source repo though, it's really wonderful, I still use it a lot for other stuff.

JH: Do you have any final thoughts to add about why you're interested in computational creative writing?

RG: I think the main thing for me is really all about pushing literature forward and creating the notion that we can move, like we might be able to move beyond linear writing in a way, and that we might be able to find new interfaces for writing that we haven't considered in the past using this technology. So I really want to see not just new interfaces but new forms and new kinds of stories. The literature that I've always respected most as a writer are the books that break the idea of what a novel can be. Roberto Bolaño I think does that really well. I just am obsessed with the idea that we can break literature in interesting ways in order to discover new kinds of stories that we can tell. That's really what I'm after.

APPENDIX C

The following is a minorly-edited transcript of a casual Zoom conversation between Judy Heflin and David Jhave Johston held on March 24, 2020.

EXCERPTS FROM CONVERSATION WITH DAVID JHAVE JOHNSTON // MARCH 24, 2020

Judy Heflin: Can you describe your broader creative practice, especially as it relates to the intersection of computation and writing? In general, what sorts of questions are you interested in and what do you find inspiring about using computational tools to explore text creatively?

David Jhave Johnston: I came from a different landscape than academia. I dropped out of University when I was 21 and then just kind of migrated around the art world for a while. I ended up writing for theatre, becoming a painter, I did a lot of installation art, didn't make much money at it, then went back to school. And as I went back to school, it was precipitated by my discovery of the internet that had just kind of emerged. Like I kind of found it around 1997, and at that point I immediately began building experiments using GIF-building software and other kinds of [...] Just creatively taking an artistic practice and moving it into a software ecosystem expands the repertoire of potentialities in ways that I found invigorating and also stimulating, and reawoke a dormant [...] it recatalyzed me. And then it just continued. You know, I did years and years of school, I did the whole ten year route, I went to high school again for a year and a half and then maths and then I took computer science and then I took a Master's in Interactive Art and then I got a PhD and all of that because in Canada you can get funding for it. So I was kind of riding partially the wave of the initial excitement that I experienced upon finding the internet and the new kinds of tools of like macromedia Flash, which allowed for a wide range of experimentation with textuality as an animated presence that evolved away from the lineated page toward a curved, topological, sensual, organic space that could change and mutate and be reactive. All of that just fascinated me, so for years I think all that I've done really, it's almost contrary to the conventional articulated theoretical modality of saying we're going to look at the materiality of the medium, because I came out of a non-academic artistic practice that had already been there for fifteen years or so. I was more approaching it as, I'm going to play. I have very few questions. My major question is what can it do? If it can do something interesting, then what do I feel like saying with it? And then once I've said something interesting, well when do I get tired with it and what's next? Those are not like really profound research questions but they're sufficient to sustain a practice. And so that's what I did for years and years. I think that's what I find inspiring about all of it.

JH: It's interesting that you were drawn to this nonlinear, non-paginated organic space. Can you talk more about exploring that and how it relates to *ReRites* as a book project that is relatively static? Is that contradiction something you had in mind? Is it even a contradiction?

DJJ: Yeah sure I think most artistic processes encompass contradictions. I certainly didn't expect to do *ReRites* [...] *ReRites* didn't begin as a book project, it was not born in my mind as like I'm going to write a research grant and find some research assistance and do something. It began as a series of [...] I was taking courses in Hong Kong, where I was a professor, in data science. When you take a course in data science through General Assembly in Data Science, which is basically like a corporate training nexus, the only place I could find where I could take a fast, accelerated, kind of hands on practical how to use data science how to think about the new methods of neural networks and machine learning and how to integrate that into practice. And so I was sitting there with a bunch of like demographic technicians from bubblegum corporations or financial traders who were all trying to optimize the monetary flow of their various networks and I was kind of interested in how do I mess with language? And that's how it began, just messing around. It's unfortunately just sort of the same haphazard story as almost everything else. I just kind of began playing with it. And then after a period of time, I began to realize that most of it wasn't creative. Like if you stare at that stuff for six months or eight months, it's fascinating on first impact how much it replicates the idioms, and structures, and nuances of conventional creative language, but then how often it falls apart. I used to call it like a six-year-old savant, you know, no life experience but with a vocabulary of like, a post-doc. It would be extraordinarily vivid but very very useless. So that's when I began editing it. And after editing it for a little while I thought I've always liked Tehching Hsieh who Nick mentions in the essay he wrote about this project, which was very appropriate. Like those one-year performance projects. Those are some of the most fundamental embodied artistic practices in the 20th century, so due to that I've adopted this idea. I've done a bunch of one-year projects kind of quietly in the background of my life, and so I set it as a goal to write every day for a year, for two hours, and it just happened organically. And it was all going along nicely and I thought well, I might as well put them into months and then maybe each month should be a book, and then I met the publishers around the corner from me, who were introduced to me by someone who I taught a workshop in Toronto. And they said what are you doing and I said I'm doing this and they said well you can publish that and I said oh wonderful. So there was no contract signed, there was no plan, there was no structure. It was in fact no research. There was kind of a flow, a coherent pressure of the technological ecosystem, in which I just swim along. In the last four months, or the last three months, three and a half months, I realized I could videotape the editing processes, and then that led to the next phase of it where it became this performative writing project. It's all just kind of, I wish I could say I was more prophetic and less instinctual but it was very instinctual.

JH: At what point did it occur to you that you wanted to print these as books? Did the month itself to you occur as a unit of the book?

DJJ: It had begun to come into my mind that I would like to publish them and by serendipity I kind of met them. And there's something else that happens when you work in [...] I've worked with web-based stuff, and installation, I did a little bit of installation, but more just pure web, and when you do that you're kind of removed from the gratifying solidity of a physical object. And also the continuum of practice where, you know people in book culture, they respect books. Net art, you know the idea of a digital poet has no meaning to 99.9% of conventional authors. So perhaps I was just trying to reclaim a bit of you know, um, respect in some sense. Right?

JH: Did you have much experience in publishing? Did you notice any difference in reception?

DJJ: I don't have much experience publishing, because I was underground. Like I published a magazine in Toronto between 1990 and 1999. I have no copies of that magazine. It was not listed anywhere, it was made on a Xerox machine by a cluster of friends, but we loved it, we made it over and over and over. So I never attempted to publish any of my works. The internet came along at precisely the right moment for me where I could say if I write something today or this week or this month, I can publish it right now myself and I had my own website by 1999. So I was just happy to work outside of the modality of conventional publishing. Also because what I was making was hybrid, and it couldn't be published anyway. If you press an interactive animation into a book, you constrain it, and lose, and reduce it in ways that are kind of impermissible to its character. So I was happy to ignore that world. But then I was really happy to have this physical object. It's amazing how gratifying it was to just see something that was and and just say oh yea, it's got like form. It's got substance. It's got cognitive validity. It's strange. Like if you look at the size of a file online and it says it's like two gigabytes you think that's amazing but fundamentally it's not the same as holding a book.

JH: Upon publishing a physical book, do you feel like it's gotten more respect because of its existence as an object? Does it change the way you think about that work?

DJJ: Yes. Because that book has been collected by six different rare books libraries now. Maybe six or seven, I'm not up on how many have been bought, but those were bought by people who would not and never will buy the collected works of my web-based net art. Because there's no [...] within those institutions there's no, even a funding parameter for how to situate like a USB key. Like what funding streams do we have? How do we justify the price for it? You know, who can judge it? I suspect it may be done in particular cases but very rarely.

JH: You said before that putting something that is more animated or web-based into a printed page would flatten it and take away the affordances of that medium. Was that something you were thinking about once you realized this was going to be a book project? Did you find anything interesting or surprising about writing using computation for the book specifically? How did you think about the medium of the book, if you did?

DJJ: I did. I suspect you're really well aware of like in the 1960's Margaret Masterman and the Haikus of the cybernetic serendipity exhibit. Basically they used a template and a replacement algorithm and an array of words. And that's been pretty much it for a long time. You can make that more and more complex, but ultimately, once you're attuned to know how that works, I could look at a piece of computer-generated text and I could say I understand that algorithm. I just have to watch it long enough to know kind of what it's doing behind the scenes because I'd written so much of it. So when the neural nets came along and I began [...] so two things happened simultaneously, Flash disappeared, so the capacity to do easy, quick animation died or dissipated. And at the same time I was kind of tired. Because I'd been doing it for years and I'd played out. So now there's a different mode of working that can happen. And in that pure language world [...] neural networks are so strong at creating language that it very swiftly became a pure writing project. And the pure writing was like coming home to the kind of writing that I used to do before the internet. And that's when I found like, the kind of seeds of why I had become a writer. Because genuinely early on I was really interested in becoming a poet as a young person, that was my goal. So I did it for years without trying to really publish it. I was doing the long road, which was, you write and write and write and there's just supposed to be a change in your soul or your artistic persona evolves or something. And I do think that kind of happened. But then along came the neural nets and suddenly it was so satisfying to sit down with this pure text, no code. I would code it at the beginning of every month, I would change the corpus at the beginning of every month, I would re-tune the code or I would find new libraries or change the structure of the parameters and output a new batch in a slightly different style. So every month I was given this opportunity to kind of explore a slightly different modality of how that might emerge, like what it might mean to write with this augmented assistance. Thoroughly enjoyable, highly recommend it.

JH: 2017, 2018 when you were writing this was when so many new frameworks and tools were emerging. Was that a big part of your process to see, every month, what new thing you could use and put it in there, almost like an archive of that year's intense growth in neural net tools?

DJJ: Yes. Because TensorFlow had come out. PyTorch was in beta. AWS got picked up by, this stands for average weights stochastic descent or something, then the PyTorch modules got picked up by SalesForces. So I just followed those tendrils of those libraries as they evolved through the year, and it just became something.

JH: What about for the corpora every month. How much did you change it? What did you use? Were you trying to go for different themes or tones related to anything?

DJJ: What I was basically after was something that replicated 20th century avant-garde conceptual poetry practice instead of like, you know, ancient Greek, or something or iambic pentameter. But then I took a lot of quotations that I continuously stored for many many years based on readings online that contain a more contemporary idiom. So the two mergers of that, with some news articles and some song lyrics, kind of create a less conventional corpora that gives a more, for me, a more vivid sense of our lived reality in the 21st century. I didn't want the computer to replicate Elizabethan English or Shakespearean prose or Herman Melville or all that stuff that's traditional. I wanted to really push it a little bit.

JH: Did you manually collect these lines?

DJJ: Automated. Automated. Automated.

JH: Just the poetry? Or also the news articles and the song lyrics?

DJJ: Everything's mildly automated. The song lyrics came from a BitTorrent back when Torrents were a thing. The citations are just grabbing lots of quotes, sending them to Tumblr, and then automatically sending them through a pipe to a document. The Poetry Foundation was, I basically took the entire site, tried to get rid of the oldest stuff, and then there's a Python library called Beautiful Soup where you run it and it just extracts whatever portion of the page that you want and throws away everything else. So I just took the poetry.

JH: So you had this initial corpus. How much would you change it every month?

DJJ: Some months not so much. I think in February I changed it kind of radically. Like I would go out and forage. I would try to find different kinds of language, different vocabularies that would maintain my interest for a year. I've never done it again. I woke up so early for an entire year and then I was just like OK that's enough. Maybe I'll return to it.

JH: What interests you about the creative uses of artificial intelligence? How do you define AI and what do you think is unique about AI as opposed to other computational methods for text generation?

DJJ: For me, it's appropriate to distinguish between the different modes of AI. Like there's been programmatic expert systems, there's Bayesian forms of learning, but it's the kind of birthplace and the resurgence of the unsupervised neural networks, which I just found it astonishing to discover that no longer did you need to implant into the system what they're now calling a dense representation of semiotic structure, or whatever it is. It really does have the syntax and all of the regulatory apparatuses that previously would need to be hard-coded in some sort of laborious way, which would never catch the edge cases, and the edge cases are kind of where poetry happens. So, I suspect that we're going to need a bunch of different words to differentiate between the errors and the changes and the epochs of AI as it shifts. Like the neural networks now, they still feel like they're really rudimentary. Like I suspect twenty years from now we'll be like, "No that stuff was so worthless." Because it really is kind of worthless. But it's still the best kind of worthless that we have so far. That's what I find fascinating about it, is that it's really doing something that is absolutely unprecedented. In the same way that we're in multiple unprecedented times, viral times, biosynthetic times, robotic war times. There's a lot of unprecedented activity going on and that's just one aspect of it. And I do think it's important for poets and artists to really engage with the toolsets that exist that are usually used by corporations for profit-driven margins or by entertainment to simply stimulate. I think there's a necessity to sort of pull literature forward into a future that it maybe doesn't want to enter. I think it's a little bit resistant.

JH: You mentioned a couple already, but who are your biggest inspirations, artistically, creatively, or poetically?

DJJ: Oh do you know Kenneth Patchen?

JH: I don't.

DJJ: Now there's a renegade for you. Some of his stuff [...] If you find a book called *The Journal of Albion Moonlight*. It's a very specific taste at a specific time. There's a kind of rage in it, but it's a Pacifist rage, and it's really a rage at a world that seems inconceivable. But he also did a lot of typographic experiments where he would have full sentences across pages, like huge words, like "the" would appear just by itself on a page with a bunch of other text around it and then you flip a couple pages and it would say "the window." So like, there are different ways of reading within his books. And then Johanna Drucker, who's astonishing. And then, do you know Johannes Helden's *Evolution*? *Evolution* came out maybe in 2011 or 2014. Johannes Helden is a Scandanavian poet who engaged a programmer to write a code system, which is perturbed by meteorological data, and it generates a kind of poetry, but based on his own mind, his own vocabulary, his own poetry. Their system is really beautiful. And then visually I was very influenced by Born Magazine. Yeah, I like that stuff. Back in the day that actually looked great. Like some of it's just amazing. But it's taste-based. You know, Nick, I don't think likes glossy things like that, he’s much more rigorous, but I’m totally an aesthetic person. I like colors and I’m susceptible to trivia.

JH: I’m specifically interested in the more consumer-oriented *ReRites / Raw Output + Responses* book. Why did you choose to publish this smaller version with the essays? How and why did you choose the sixty poems that were put into it? Do you have a favorite of those? Why did you find it important to include the raw output as well?

DJJ: The book itself, the paperback [...] I’m dealing with a really small publishing company, they’re an art book group of maybe about three or four people, the core team, so it’s not a major infrastructure, and they’ll probably maybe break even on the hardcovers, in an edition of twelve. So the paperbacks are more about putting into people’s hands two things: a morsel of the poetry in an affordable format, but also I think it comes back to my core insecurities, because it has all these validating essays in it that I went out and found from people I know. So the essays kinds of say this is not just abstract crap by some random person. The essays say well we are prestigious certified people with major positions at institutions from all over North America and we endorse this. And that’s strange, like that’s a cynical way to see it, but it’s also I think a necessary step for digital poetry to bridge the credibility gap. Because in many people’s minds the kind of poetry that computers are writing, even when we use a hybrid-augmented technique, it’s insufficient. They kind of, there’s this sense that it’s tainted, or contaminated by the influx of the machine. That there must have been an insufficiency on the part of the human that required working with the machine or perhaps its insufficient because they focused too much on coding. So I think all of those thoughts kind of swirled around and I liked the idea of having this accessible small version which would have these validating essays. And the chosen poems, the 60 poems, were really to some degree trying to provide a sense of fragments from the whole set and then to have one or two of the favorites. And there’s one that begins with a set of numbers, and I’m very bad at citing my own poetry library. Why do I call it my own? I really don’t consider it my own. It’s more [...] It’s almost like my entire life has been allowing systems of pressure to sort of force me along and become a conduit. That’s the more ancient model of the Orphic muses crouched over the hallucinogenic vent. So in that sense I was just following the contours of what was presented to me by the AI. But there’s a poem in there I think from April or March where it begins with numbered sets, and those I feel like are more coherent or more joined and they have a kind of forlorn nomenclature and atmosphere that reminds me of Charles Olson or something like that. And I believed it important to include the raw output because I wanted people to be able to say yes there was human physical labor, that in some cases maybe I removed some things that were extraordinary, which is true. Because really it’s a process of carving and each individual would find a different set of poems within this mammoth output, avalanche.

JH: Yeah, a lot of the videos are really dramatic to watch because you erase things I definitely would have kept.

DJJ: It's like watching someone skiing. It's kind of like, we're presented with this stuff and you're on a gravitational path. And if you allow the momentum of that to continue then you just continue cutting and carving and so there's not a lot of contemplating time. Once I switched to the recording, something interesting about that, I've almost used it as a kind of writing technique now, because if I turn on a QuickTime screen grab and I'm writing, just writing, because now I'm writing a novel using AI, but when I know that I'm screen-grabbing, I don't stop, I don't get in these frozen moments, because I'm performing. There's somebody watching even if there's nobody watching the video. I can't go get a coffee, I can't stop. I can slow down a little bit, but I can't methodically read through what I wrote. Somehow the velocity of the system, of being observed, creates in me the compulsion to write more. But that's got nothing to do with AI, it's about voyeurism, self-voyeurism or something, but it works.

BIBLIOGRAPHY

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

Abbott, H. Porter. *The Cambridge Introduction to Narrative*. Cambridge University Press, 2008.

Armitage, Catherine. "Google Autocomplete Is Being Used to Create Poetry." The Sydney Morning Herald, February 2, 2014. https://www.smh.com.au/opinion/google-autocomplete-is-being-used-to-create-poetry-20140201-31tp4.html.

Barthes, Roland. "The Death of the Author." In *Image / Music / Text*, translated by Stephen Heath, 142–47. London: Fontana, 1977.

Black, Annetta. "Canard Digérateur De Vaucanson (Vaucanson's Digesting Duck)." Atlas Obscura, January 30, 2010. https://www.atlasobscura.com/places/canard-digerateur-de-vaucanson-vaucansons-digesting-duck.

Booten, Kyle, and Dan Rockmore. "The Anxiety of Imitation: On the 'Boringness' of Creative Turing Tests." electronic book review, May 3, 2020. https://electronicbookreview.com/essay/the-anxiety-of-imitation-on-the-boringness-of-creative-turing-tests/.

Brownlee, Jason. "Why One-Hot Encode Data in Machine Learning?" Machine Learning Mastery, April 27, 2020. https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

Byrne, Brendan C. "An Excess of Consciousness." Rhizome, July 10, 2018. https://rhizome.org/editorial/2018/jul/10/an-excess-of-consciousness-1/.

Calvino, Italo. "Cybernetics and Ghosts." In *The Uses of Literature: Essays*, 3–27. San Diego, New York, London: Harcourt Brace & Company, 1987.

Carnegie Mellon University Speech Group. "The CMU Pronouncing Dictionary." Sphinx knowledge base tools, n.d. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Cayley, John. "The Code Is Not the Text (Unless It Is the Text)." electronic book review, October 9, 2002. https://electronicbookreview.com/essay/the-code-is-not-the-text-unless-it-is-the-text/.

Chamberlain, William, and Thomas Etter. *The Policeman's Beard Is Half-Constructed: Computer Prose and Poetry*. New York, NY: Warner Software/Warner Books, 1984.

Ciocca, Sophia. “How Does Spotify Know You So Well?” Medium, October 10, 2017. https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe.

Clarke, Arthur C. “The Steam-Powered Word Processor.” In *The Collected Stories of Arthur C. Clarke*. Orb Books, 2002.

Croll, Angus. *If Hemingway Wrote JavaScript*. No Starch Press, 2014.

Dahl, Roald. *The Great Automatic Grammatizator*. Puffin, 1997.

Data Catchup. “The Difference between Artificial Intelligence, Machine Learning and Deep Learning.” May 27, 2019. https://datacatchup.com/artificial-intelligence-machine-learning-and-deep-learning/.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding,” May 24, 2019. https://arxiv.org/pdf/1810.04805.pdf.

Downer, Adam. “Harry Potter Predictive Text Chapter.” Know Your Meme, January 25, 2018. https://knowyourmeme.com/memes/harry-potter-predictive-text-chapter.

Emerson, Lori. *Reading Writing Interfaces: From the Digital to the Bookbound*. Minneapolis: University of Minnesota Press, 2014.

Erslev, Malthe Stavning. “I Forced a Bot to Read over 1,000 Papers from Open Access Journals and Then Asked It to Write a Paper of Its Own. Here Is the Result. Or, a Quasi-Materialist Approach to Bot-Mimicry.” *A Peer-Reviewed Journal About* 8, no. 1 (2019): 114–26. https://doi.org/10.7146/aprja.v8i1.115419.

Explosion. “SpaCy · Industrial-Strength Natural Language Processing in Python.” https://spacy.io/.

Firth, J. R. “A Synopsis of Linguistic Theory 1930-1955.” In *Studies in Linguistic Analysis*, 1–32. Oxford, 1962.

Flores, Leonardo. “Distant Writing.” YouTube. MITH Digital Dialogues, March 10, 2020. https://youtu.be/It2seaB5x-U.

Franniiie. “How about We Use Predictive Text to Write Our Own Epitaphs? Write ‘Here Lies [YOUR NAME]. [Preferred Pronoun] Was..." and Let Your Phone Finish the Rest by Just

Pressing the Middle Text Button." reddit, January 2, 2018. https://www.reddit.com/r/funny/comments/7nlnd7/how_about_we_use_predictive_text_to_write_our_own/.

French, Scott. *Just This Once*. New York, NY: Carol Publishing Group, 1993.

G., Brad. "Type 'I Was Born' and Then Let Your Predictive Write Your Autobiography." Twitter, November 3, 2017. https://twitter.com/therealbradg/status/926260526270652416.

Garfinkel, Simon. "IRobot Roomba." MIT Technology Review, October 9, 2002. https://www.technologyreview.com/2002/10/09/234680/irobot-roomba/.

Genette, Gérard. *Paratexts: Thresholds of Interpretation*. Cambridge University Press, 1997.

Goodwin, Ross. "Narrated Reality." https://narratedreality.com/, 2016.

Goodwin, Ross. "Adventures in Narrated Reality." Medium. Artists Machine Intelligence, June 9, 2016. https://medium.com/artists-and-machine-intelligence/adventures-in-narrated-reality-6516ff395ba3.

Goodwin, Ross. *1 The Road*. Paris: Jean Boîte Éditions, 2018.

Google Arts & Culture. "The Technology and People behind Please Feed the Lions." Google, 2018. https://artsandculture.google.com/exhibit/the-technology-and-people-behind-please-feed-the-lions/uQJy-TQgHNPkJw.

Google Arts & Culture. "The Lion's Roar: 23rd September." Google, 2018. https://artsandculture.google.com/theme/the-lion-s-roar-23rd-september/KQIytu3ZhNlDKw.

Greene, Tristan. "Burger King Trolls Artificial Intelligence with New Ads." The Next Web, September 28, 2018. https://thenextweb.com/artificial-intelligence/2018/09/28/burger-king-trolls-artificial-intelligence-with-new-ads/.

Hart, Michael. "The History and Philosophy of Project Gutenberg." Project Gutenberg, August 1992. https://www.gutenberg.org/wiki/Gutenberg:The_History_and_Philosophy_of_Project_Gutenberg_by_Michael_Hart.

Hart, Michael. "Project Gutenberg Mission Statement." Project Gutenberg, December 25, 2007.

https://www.gutenberg.org/wiki/Gutenberg:Project_Gutenberg_Mission_Statement_by_Michael_Hart.

Hartman, Charles O. *Virtual Muse: Experiments in Computer Poetry*. Wesleyan University Press, 1996.

Hayles, N. Katherine. “Translating Media: Why We Should Rethink Textuality.” *The Yale Journal of Criticism* 16, no. 2 (2003): 263–90. https://doi.org/10.1353/yale.2003.0018.

Hayles, N. Katherine. *My Mother Was a Computer: Digital Subjects and Literary Texts*. Chicago: University of Chicago Press, 2005.

Hayles, N. Katherine. “Electronic Literature: What Is It?” Electronic Literature Organization, January 2, 2007. https://eliterature.org/pad/elp.html.

Hayles, N. Katherine. “Intermediation: The Pursuit of a Vision.” *New Literary History* 38, no. 1 (2007): 99–125. https://doi.org/10.1353/nlh.2007.0021.

Henrickson, Leah. “Towards a New Sociology of the Text: the Hermeneutics of Algorithmic Authorship,” 2019.

Honnibal, Matthew. “Embed, Encode, Attend, Predict: The New Deep Learning Formula for State-of-the-Art NLP Models.” Explosion, November 10, 2016. https://explosion.ai/blog/deep-learning-formula-nlp.

Jean Boîte Éditions. “A Gonzo AI Takes the Road after Kerouac to Give Us the First Novel Written by a Machine.” Press release, September 2018. http://www.desrimais.com/FTP/JBE/00-CP_1_the_Road_EN.pdf.

Jean Boîte Éditions . “Art Machines.” JBE Books, n.d. https://www.jbe-books.com/collections/art-machines.

Johnson, Justin. “Jcjohnson/Torch-Rnn.” GitHub, February 16, 2016. https://github.com/jcjohnson/torch-rnn/blob/master/doc/modules.md.

Johnson, Justin. “Jcjohnson/Densecap.” GitHub, June 13, 2017. https://github.com/jcjohnson/densecap.

Johnston, David Jhave. “RERITES : Machine Learning Poetry Edited by a Human.” Glia, n.d. http://glia.ca/rerites/.

Johnston, David Jhave. *Aesthetic Animism: Digital Poetry's Ontological Implications*. Cambridge, MA: The MIT Press, 2016.

Johnston, David Jhave. “Saved Output.” GitHub, April 19, 2017. https://github.com/jhave/pytorch-poetry-generation/blob/master/word_language_model/GENERATED/Terminal Saved Output-19042017-1331-1MCorpus-2h48m.txt.

Johnston, David Jhave. “Jhave/RERITES-AvgWeightDescentLSTM-PoetryGeneration.” GitHub, June 30, 2018. https://github.com/jhave/RERITES-AvgWeightDescentLSTM-PoetryGeneration.

Johnston, David Jhave, Allison Parrish, Johanna Drucker, Kyle Booten, John Cayley, Lai-Tze Fan, Nick Montfort, Mairéad Byrne, and Chris Funkhouser. *ReRites - Raw Output / Responses*. Edited by Stephanie Strickland. Quebec: Anteism Books, 2019.

Karpathy, Andrej. “The Unreasonable Effectiveness of Recurrent Neural Networks.” The Unreasonable Effectiveness of Recurrent Neural Networks, May 21, 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Kelleher, John D. In *Deep Learning (The MIT Press Essential Knowledge Series)*. Cambridge, MA: MIT Press, 2019.

Kim, Young Hee. “David Markson’s This Is Not a Novel: A Deviation from Traditional Fiction Writing.” *Studies in English Language & Literature* 43, no. 2 (2017): 27–47. https://doi.org/10.21559/aellk.2017.43.2.002.

Kosner, Anthony Wing. “Google Cabs And Uber Bots Will Challenge Jobs 'Below The API'.” Forbes, February 4, 2015. https://www.forbes.com/sites/anthonykosner/2015/02/04/google-cabs-and-uber-bots-will-challenge-jobs-below-the-api/#403bfce569cc.

Läufer Milton. *A Noise Such as a Man Might Make*. Denver: Counterpath, 2018.

Leiber, Fritz. *The Silver Eggheads*. Ballantine Books, 1979.

Ludipe. “Type ‘You Are Not a Real Gamer until You’ and Let Your Predictive Text Finish That Quote for You.” Twitter, April 4, 2019. https://twitter.com/Ludipe/status/1113717316536209408.

Malkiel, Burton. *A Random Walk Down Wall Street*. Norton, 1973.

Manjavacas, Enrique, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont. “Synthetic Literature: Writing Science Fiction in a Co-Creative Process.” *Proceedings of the Workshop on Computational Creativity in Natural Language Generation*, 2017. https://doi.org/10.18653/v1/w17-3904.

Martin, Nicky, and Nat Towsen. “Tutorial.” Botnik, n.d. https://botnik.org/tutorial.html.

McCulloch, Gretchen. “Autocomplete Presents the Best Version of You.” Wired, February 9, 2019. www.wired.com/story/autocomplete-presents-the-best-version-of-you/.

Milman, Henry Hart. “Nala and Damayanti and Other Poems.” Edited by Sankar Viswanathan. Project Gutenberg, October 12, 2006. http://www.gutenberg.org/files/19529/19529-0.txt.

Montfort, Nick. “The Cut-Up Method of Brion Gysin.” In *The New Media Reader*, 89–91. Cambridge, MA: MIT Press, 2003.

Montfort, Nick. “Conceptual Computing and Digital Writing.” In *Postscript: Writing After Conceptual Art*, edited by Andrea Andersson, 197–210. Toronto: University of Toronto Press, 2018.

Montfort, Nick. “Computer-Generated Books.” Post Position, October 31, 2019. https://nickm.com/post/2016/07/computer-generated-books/comment-page-1/.

Montfort, Nick. “Nano-NaNoGenMo or #NNNGM.” Post Position, November 16, 2019. https://nickm.com/post/2019/11/nano-nanogenmo-or-nnngm/.

Montfort, Nick, and Judy Heflin. “Computer-Generated Literature, Author/Programmers, and Machine Voices.” In *The Cambridge Companion to Literature in the Digital Age*. Cambridge University Press, n.d..

O'Brien, Chris. “Meet the Japanese Artist Who Made the World's Largest GPS Drawing for His Marriage Proposal.” VentureBeat, November 20, 2014. https://venturebeat.com/2014/11/17/meet-the-japanese-artist-who-made-the-worlds-largest-gps-drawing-for-his-marriage-proposal/.

Orwell, George. *1984*. Signet Classics , 1950.

Parrish, Allison. “Understanding Word Vectors: A Tutorial for ‘Reading and Writing Electronic Text," a Class I Teach at ITP.” GitHub, n.d. https://gist.github.com/aparrish/2f562e3737544cf29aaf1af30362f469.

Parrish, Allison. “Poetic Sound Similarity Vectors Using Phonetic Features.” *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, September 19, 2017.

Parrish, Allison. “Experimental Creative Writing with the Vectorized Word.” YouTube. Strange Loop, September 30, 2017. https://www.youtube.com/watch?v=L3D0JEA1Jdc.

Parrish, Allison. *Articulations*. Denver, CO: Counterpath, 2018.

Parrish, Allison. “Gutenberg, Dammit.” GitHub, August 2018. https://github.com/aparrish/gutenberg-dammit.

Parrish, Allison. “A Gutenberg Poetry Corpus.” GitHub, August 2018. https://github.com/aparrish/gutenberg-poetry-corpus.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation, 2014. https://nlp.stanford.edu/projects/glove/.

Pérez y Pérez Rafael. *Mexica: 20 Years-20 Stories [20 años-20 Historias]*. Denver: Counterpath, 2017.

Pold, Søren, and Malthe Stavning Erslev. “Data-Realism: Reading and Writing Datafied Text.” electronic book review, March 5, 2020. https://electronicbookreview.com/essay/data-realism-reading-and-writing-datafied-text/.

Poole, Steven. “The Rise of Robot Authors: Is the Writing on the Wall for Human Novelists?” The Guardian. Guardian News and Media, March 25, 2019. https://www.theguardian.com/books/2019/mar/25/the-rise-of-robot-authors-is-the-writing-on-the-wall-for-human-novelists.

Pozin, Ilya. “20 Entrepreneurs Shaking Up New York's Tech Scene.” Forbes, September 15, 2014. https://www.forbes.com/sites/ilyapozin/2014/09/13/20-entrepreneurs-shaking-up-new-yorks-tech-scene/#1078c57029e7.

Pressman, Jessica. “House of Leaves : Reading the Networked Novel.” *Studies in American Fiction* 34, no. 1 (2006): 107–28. https://doi.org/10.1353/saf.2006.0015.

Queneau, Raymond. *One Hundred Million Million Poems*. Kickshaws, 1983.

Radford, Alec, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. “Better Language Models and Their Implications.” OpenAI, February 14, 2019. https://openai.com/blog/better-language-models/.

Rettberg, Scott. *Electronic Literature*. Cambridge, UK: Polity Press, 2019.

Rimbaud, Arthur. “Youth.” PoemHunter.com, April 3, 2010. https://www.poemhunter.com/poem/youth-5/.

Ruder, Sebastian. “On Word Embeddings - Part 1.” Sebastian Ruder, October 6, 2019. https://ruder.io/word-embeddings-1/.

Salton, Gerard. “Advanced Information-Retrieval Models: The Vector Space Model.” In *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer.* , 313–19. Addison-Wesley , 1989.

Schimkowitz, Matt. “I Forced a Bot.” Know Your Meme, June 15, 2018. https://knowyourmeme.com/memes/i-forced-a-bot.

Shillingsburg, Peter. *From Gutenberg to Google: Electronic Representations of Literary Texts*. Cambridge University Press, 2006.

Shmyrev, Nickolay. “CMUSphinx Open Source Speech Recognition.” CMUSphinx Open Source Speech Recognition. Carnegie Mellon University, n.d. https://cmusphinx.github.io/.

Steyerl, Hito. “A Sea of Data: Pattern Recognition and Corporate Animism (Forked Version).” In *Pattern Discrimination*, 1–22. University of Minnesota Press, 2018.

Tang, Jian, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. “Context-Aware Natural Language Generation with Recurrent Neural Networks.” Cornell University, November 29, 2016. https://arxiv.org/abs/1611.09900.

Taper. “Taper: an Online Literary Magazine for Small Computational Pieces.” Bad Quarto. https://taper.badquar.to/.

Torch. “Scientific Computing for LuaJIT.” http://torch.ch/.

Vincent, James. “Instagram Explains How It Uses AI to Choose Content for Your Explore Tab.” The Verge, November 25, 2019. https://www.theverge.com/2019/11/25/20977734/instagram-ai-algorithm-explore-tab-machine-learning-method.