

Meaning-Making and Randomization in E-Poetry Machines

Amanda Hill, St. Mary's University

Laura Okkema, University of Central Florida

In this paper, the authors propose a dialectic of pattern and randomness as an interpretative lens for electronic poetry. By examining the complex interplay of recurring patterns and randomly generated elements in two electronic poems, they demonstrate how poetry and code work together and how meaning can be made from randomly generated poetic works. They showcase two of their own electronic poetry projects as case studies, the poem *Wayfarer's Song* (Okkema) and the *Dada Poetry Generator* (Hill). *Wayfarer's Song* generates a Villanelle poem from a set of randomly chosen, pre-written verses. On the level of content, the poem exemplifies the interplay of pattern and randomness by juxtaposing arbitrary arrangement with the poetic pattern of Villanelle. Reading the poem in light of pattern and randomness reveals further layers of meaning on the levels of code, interface, and hardware. The *Dada Poetry Generator* engages users in creating poems from "found" texts. Because each iteration of the machine generates a different arrangement of the texts, the context and meaning of the poems change each time the code runs. In deforming and decontextualizing these texts, the users encounter symbolic randomness. Examining the interplay of pattern and randomness in each project reveals the multifaceted and complex meanings which can emerge from readings of electronic poetry.

In this paper we use a dialectic of the concepts 'pattern' and 'randomness' as a lens to interpret electronic poetry (e-poetry) and explore how meaning can be made from randomly generated poetic works. We present two e-poetry project case studies: the poem "*Wayfarer's Song*," composed and coded by Laura Okkema, and the *Dada Poetry Generator* created by Amanda Hill. By identifying moments of pattern and randomness in each e-poem, we showcase a method of reading and interpreting works of digital literature in their respective contexts. Case One examines "*Wayfarer's Song*" as an example of digital Oulipo poetry, exploring the interplay of pattern and randomness at the levels of content, code, interface, and hardware. Case Two presents the *Dada Poetry Generator*, where users, in deforming and decontextualizing 'found' texts, encounter symbolic randomness as an opportunity for further exploration and meaning-making. Our goal is to demonstrate that the pat-

tern-randomness dialectic is a useful tool in understanding how meaning emerges on several levels in works of electronic poetry.

We adapt the notion of a pattern-randomness dialectic from the work of Katherine Hayles. Hayles first introduced the pattern-randomness dialectic two decades ago in *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics* (1999), where she examines how electronic media have changed signification. Hayles explores phenomena which arose in the 1980s and 90s, such as the replacement of the physical desk with a virtual desktop, the possibility of virtual reality, or the exchange of the typewriter (which left physical, tangible marks on the page) for the keyboard, which yields virtual letters made visible through countless chains of code unknown to the user. Text, once stable and printed on a page, becomes a flickering image, with several layers of commands and human-machine communication between the keystroke and the letter on the screen. Hayles expresses concern over what she then perceived as a trend towards dematerialization, arguing that the rise of information technology in the 20th century has led to a disregard for the material, physical, components of the human (and textual) body. Following Hayles, the flickering text is characterized by a complex dialectic of pattern and randomness, expressed in the many layers of machine-computer language that lie between signifier and signified.

In later publications (2005, 2008), Hayles acknowledges that debates over a rigid binary between materiality and immateriality have begun to fade in the 21st century, in favor of more complex discussions concerning the entanglement of bodies, machines, materiality, virtuality, code, text, information and meaning. Her recent scholarship in fact revolves around electronic literature and thus addresses electronic poetry, but she does not apply the expression 'pattern-randomness dialectic' from her earlier work to this subject. Here, we rely on Hayles' argument that pattern and randomness are two concepts which are central to understanding digital media and information technology, while adapting this dialectic as a lens to interpret electronic poetry. As the case studies will demonstrate, identifying moments of pattern and randomness allows readers and students of electronic poetry to explore the many facets and layers of meaning of these works.

In the context of this paper, we engage the term 'pattern' to mean a recognizable structure, i.e. any configuration or arrangement of elements (whether it be particles, molecules, building bricks, knitted fabric, words, sentence structures etc.) which can be identified as such by a human or machine. By randomness, we mean the arbitrary determination or disruption of elements within a structure. While this definition is somewhat narrow (excluding the idea that randomness can exist apart from structure altogether), it is helpful in the context of electronic literature, as it stresses the idea that the sequence

of words in a text can be determined haphazardly, for example via a randomized algorithm. Moreover, defining randomness in this fashion allows us to discuss the occurrence of unpredictable events in otherwise regulated processes, such as glitches and errors in a computer program.

We use the term “electronic poetry” or “e-poetry” following e-poet Stephanie Strickland’s (2009) definition: “E-poetry relies on code for its creation, preservation, and display: there is no way to experience a work of e-literature unless a computer is running it.” Whenever we use the term “e-poetry machine,” we specifically refer to such e-poems which rely on a randomized algorithm; this includes both of our case studies.

A Brief History of (E-)Poetry Machines

Poetry machines are not unique to electronic media; the precursor to the e-poetry machine is the *Oulipo* poem. The *Ouvroir de littérature potentielle* (*Oulipo*), or workshop of potential literature, was a group of avant-garde writers and mathematicians founded in the 1960s. Raymond Queneau’s (1961) poem “Cent Mille Millions de Poèmes,” for instance, is a literary machine with the potential to produce $10^{14} = 100,000,000,000,000$ different sonnets. Queneau crafted 10 sonnets in such a manner that each verse could be replaced with another verse occupying the same line-position within the structure of the sonnet. The work as a whole is unreadable to humans; to arrive at an understanding of the work, the underlying mathematical structure must be taken into account.

Works of 20th century avant-garde poetry can help us better understand e-poems because they often rely on ‘analog’ mathematical algorithms, thus forming precursors to the computer algorithms in e-poetry. Digital poet and scholar Loss Pequeño Glazier (2008) argues that e-poetry is an heir to 20th century avant-garde poetry and its innovative poetic practice, including the works of William Carlos Williams, Ezra Pound, as well as Raymond Queneau and François Le Lionnais of *Oulipo*. Glazier makes the case that poetry as practiced by *Oulipo* is “uniquely qualified to equip us to enter and investigate the expanded textuality of digital writing” (p. 121). In emphasizing the ‘expanded’ textuality of electronic writing, Glazier points to the importance of interpreting e-poetry in the context of digital media: e-poems rely on the unique affordances of the electronic medium to make meaning, and therefore cannot simply be interpreted with the same tools and methods which are traditionally applied to print poetry.

Contemporary works of generative poetry borrow their structure from “Cent Mille Millions de Poèmes;” however, the literal cutting and pasting that Queneau intended for his poem is translated into a machinic algorithm

in e-poetry machines. To produce an iteration of a poem, print-based poetry machines required that a human reader physically manipulate pages of a book. In e-poetry machines, this process is replaced through the execution of the code, or *performance* of the poem. Scott Rettberg (2014) affirms that “[w]orks of e-lit are generally interconnected in ways that are not easily amenable to print publication, and they branch, or importantly perform on request” (p. 170). Performance occurs when the e-poem’s code is executed on a compatible electronic device. For example, Nick Montfort’s (2009) e-poem “Taroko Gorge” performs by randomizing verses based on a pre-coded grammatical pattern, drawing on a set of arrays containing substitutable verbs, nouns, and adjectives. Another, more expansive example for generative poetry is Scott Rettberg’s (2009) Ruby-based poetry machine “Frequency.” Rettberg applies the principle of Queneau’s machinic algorithm but takes the idea of potential poetry far beyond the single form of the sonnet. Taking advantage of the affordances provided by digital media, *Frequency* can generate poetry based on various constraints, outputting forms such as Shakespearian, Petrarchan, and Spenserian sonnets, as well as Haiku and Tanka.

The successful execution of an e-poem’s code depends on a variety of factors, such as the availability of compatible software. Depending on those factors, performance may not always be smooth; however, in e-poetry, randomness, disruptions, glitches and noise are, in fact, desirable, and sometimes build intentionally into the poem. Lori Emerson (2014) argues that this “glitch aesthetic” radically redefines “what counts as an aesthetic object or an aesthetic experience and asserts that its disruptiveness (in that a glitch constitutes a moment of dysfunctionality in the computer system) defamiliarizes the slick surface of the hardware/ software of the computer and so ideally transforms us into critically minded observers of the underlying workings of the computer.” In other words, electronic works are often built to complicate the interface, reminding us that the screen is merely the surface level of meaning-making in electronic communication. While the technology industry may work hard to draw our attention to the shiny interface of their product, away from the levels of code and hardware, e-poetry and literature push in the opposite direction. The two cases below serve to demonstrate how the concepts of pattern and randomness can help us strip and analyze the different layers of meaning in e-poetry machines.

Case One: “Wayfarer’s Song”

“Wayfarer’s Song” (<http://scriboergosum.net/test.php>) generates a Villanelle poem from a set of randomly chosen, pre-written verses. Readers who visit the site that hosts “Wayfarer’s Song” see the poem appear one line after the other, at an interval of a half second. Five seconds after the poem is complete,

the site automatically refreshes, the old version is lost, and a new version of the poem appears one line at a time. Figure 1 shows three screenshots of “Wayfarer’s Song” during performance.

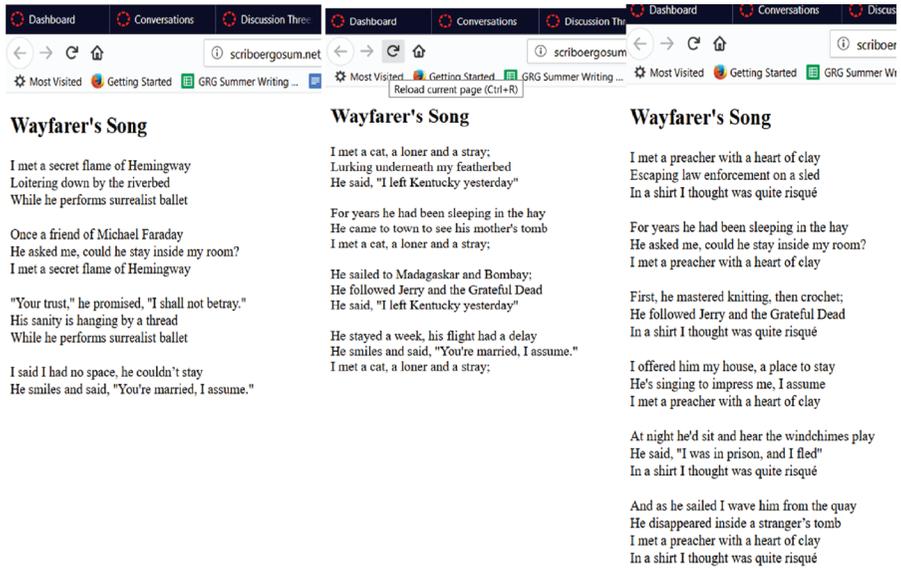


Figure 1. Three iterations of “Wayfarer’s Song”

“Wayfarer’s Song” is essentially an electronic *Oulipo* poem. Like Montfort’s “Taroko Gorge,” “Wayfarer’s Song” was built in html and JavaScript and thus relies on a similar randomization algorithm. It is an *Oulipo* poem in the sense that it emulates the principle which underlies Queneau’s “Cent Mille Millions de Poèmes” in a digital environment, but instead of sonnets, it produces Villanelle poems. To interpret works like “Wayfarer’s Song,” “Taroko Gorge,” or “Frequency,” it is important to bear in mind that an individual instantiation – such as the screenshot of a rendering of “Wayfarer’s Song,” or a single Haiku plotted by the *Frequency* machine – is only a fragment of the entire work. Looking at one rendering can be insightful in many ways, as will become evident in the discussion of The Dada Poetry Generator, but it cannot lead to a thorough understanding of the complete work.

Poems like “Wayfarer’s Song” function on several levels of meaning; a lens of pattern and randomness allows us to explore each level individually. On the level of content, we find patterns of literary form - the Villanelle structure in the case of “Wayfarer’s Song.” As a form, the Villanelle follows a strict pattern of five tercets and one quatrain; the first and third line are alternately repeated at the end of each tercet, and finally brought together in a couplet at the end. The rhyme scheme is accordingly ABA ACA ACAA. In the case of

“Wayfarer’s Song,” the Villanelle pattern serves as a container for randomized content. This brings us to the level of code.

The program-part of “Wayfarer’s Song” draws the content of each line from a set of pre-composed verses. A villanelle has nineteen lines, six of which are repeated; this means thirteen lines are unique. In “Wayfarer’s Song,” five alternative versions exist for each of the thirteen unique lines. When the program loops through an iteration, it is told: randomly pick a line from the array called “Line A,” i.e. the set of lines that grammatically and rhythmically fits the first line of the Villanelle and print it. The program then moves on to line 2, and so on. Figure 2 shows part of the poem’s code, including the arrays containing the lines and the randomization function.

```
<script type="text/javascript">

var line = [];
var lineA = ["I met a cat, a loner and a stray;", "I met a creature, beak and feathers grey", "I met a preac"];
var lineB = ["Lurking underneath my featherbed", "Loitering down by the riverbed", "In my kitchen, stealing"];
var lineC = ["As he was on the road to Mandalay", "He said, \"I left Kentucky yesterday\"", "In a shirt I th"];
var lineD = ["He was known a cannibal gourmet", "For years he had been sleeping in the hay", "Once he was a j"];
var lineE = ["He asked me, could he stay inside my room?", "He came to town to see his mother's tomb", "He sa"];
var lineF = ["First, he mastered knitting, then crochet:", "He sailed to Madagascar and Bombay:", "He stole"];
var lineG = ["A questionmark emerges from his head", "He followed Jerry and the Grateful Dead", "He said, \""];
var lineH = ["I offered him my house, a place to stay", "He stayed a week, his flight had a delay", "He took"];
var lineI = ["He took my wrist and sniffed for my perfume", "And from his flaring nostrils rises fume", "He"];
var lineJ = ["His knees caved in when he began to pray", "He chokes and coughs the ashes from the tray", "At"];
var lineK = ["He said, \"I was in prison, and I fled\"", "A river from his eye is running red", "No gentlem"];
var lineL = ["And as he sailed I wave him from the quay", "He danced and sung all night to my dismay", "And"];
var lineM = ["And so he leaves at dawn to meet his doom", "He left at dawn and journeyed to Khartoum", "He j"];

function randomize(line) {
    var rand = line[Math.floor(Math.random()*line.length)];
    return rand;
};
```

Figure 2. A section of code from “Wayfarer’s Song”

While coding the poem, the programmer relies on patterns and rules of the respective programming language (JavaScript in our example). These patterns must be strictly followed so the machine can process the commands. For the computer, the ‘human’ content of the poem – for example, the word ‘wayfarer’ – is just a random string of signs to which meaning is assigned via variables. For humans, machine language –especially on the deeper levels, such as binary code– may seem like a random assembly of numbers. In other words, when composing a work like “Wayfarer’s Song,” the programmer-author must keep two audiences in mind: human *and* machine.

Part of the aesthetic in print poems is their arrangement on the page: the line breaks and verses may form patterns or, in the case of concrete poetry, specific visual forms and images. In e-poetry, on the level of the visual interface, we can identify similar visual patterns. In addition, however, e-poems are embedded in the environment of the screen, the operating system, and the browser in which they are run. We may, for example, identify familiar design patterns when we look at the poem displayed in our preferred browser window. If viewed on an old screen, we may see flickering patterns of color and light while reading the poem, whereas a newer screen will provide smooth,

even backlight. These visual patterns can be disrupted by random errors and glitches on the display, which may influence our ability to read the poem.

To appreciate works of e-poetry in their full complexity, we need to look beyond the surface level of the interface and consider the deepest level of the machine, the hard drive, where the poem is physically inscribed. We often think of screen text as effervescent and immaterial, which is far from true. As Matthew Kirschenbaum (2012) has pointed out, hard drives are “mechanisms of extreme inscription” (p. 74). Interpretation on the microscopic level of hardware is unfamiliar territory for most humanities scholars. Again, looking through the lens of pattern and randomness can guide our interpretation. During inscription, patterns of binary values are converted to voltage spikes which pass through the read/write head of the drive. Randomness and noise can enter the system on this level as well – even though the user will hardly notice, except maybe in extreme cases – for example when lightning literally strikes. While this is a radically simplified description of a highly complex process, it reminds us of the complexity of electronic works and the importance of looking beyond the interface in our readings of such texts.

Interpreting “Wayfarer’s Song” with the pattern-randomness dialectic in mind demonstrates that e-poems are not so much about content as they are about the medium through which they are delivered: they comment on its nature, its materiality, its textuality. Like avant-garde poems, they might be called ‘meta’ poems or, as Oulipo has called them, ‘potential’ poems. The following section introduces another example, The Dada Poetry Generator, and expands upon ways in which the pattern-randomness dialectic can serve as a lens to make meaning from randomly generated digital texts.

Case Two: The Dada Poetry Generator

The Dada Poetry Generator (<https://bit.ly/2HL9CO5>) is an e-poetry machine that engages users in creating a poem from several “found” texts: a news article, a passage from a book, and an excerpt from a website. It invites readers to make new inferences about the texts which are currently in front of them. Because each iteration of the machine will generate a different arrangement of the texts, the context and meaning of the poem can change each time the code runs. In deforming and decontextualizing these texts, users will encounter symbolic randomness. This seeming nonsense is an opportunity for further exploration and meaning-making. The three texts a user chooses will relate to each other in different ways. If the texts the reader chooses cover different topics or come from different realms of the reader’s life, the Dada Poetry Generator additionally provides a way to make a connection to various branches of daily life (eg. home, work, school) which create the user’s “world of experi-

ence.” O’Gorman suggests nonsense “can take us across cultural and cognitive fields.” By connecting texts from different areas of our lives, we make leaps from one subject to another and are afforded the opportunity to find common themes and patterns that are emerging in our daily lives and our society.

Visitors to the Dada Poetry Generator can begin a new poem by clicking the “Make a New Poem” link in the center of the page. Visitors are transported from the home page to the Dada Poetry Generator. On this page, they will find three spaces to copy and paste specific texts into the generator. The machine asks for texts culled from a news article, a book, and a blog or website to encourage users to engage with texts they are currently exploring in their daily lives in new ways. From each of those texts, visitors select one important word from which to create a title for their new poem. Visitors then sign their name and click “submit.” When visitors submit their responses, they are redirected to a page that shows a poem created out of the three texts they provided. Figure 3 gives an example of such poems.

Cooperate Seeing Favorite

**their shop was pleasant to eat in, clean and brightly lit by a b
sight, as if girls at large in the Naval Yard might scatter li
yesterday, I answered questions under oath about every top
ey have requested and will continue to cooperate."It all started with
e judge's background check, following a deal struck by Senat
judge's background check, following a deal struck by Senate Re
week to look into into allegations of sexual assault broug
ll stools where they sat measuring all day. Anna sensed anxi
cess. "Throughout this process, I've been interviewed by
lon? No one knew, and Mr. Voss was not a man who welcomed
o is near you. Would you like to save it as your favorite?
unch over the disapproval of her supervisor, Mr. Voss, who liked the
that might affect the measurements she and the other girls too
might affect the measurements she and the other girls took—or was
hot September days when Anna first came to work there. Now she
a bank of second-story windows. It had conditioned air, a
their shop was pleasant to eat in, clean and brightly lit by a b
o save it as your favorite?
in order to function? No one knew, and Mr. Voss was not
BI to conduct a limited "supplemental investigation" in
ckground check, following a deal struck by Senate Republicans to mov
oved ahead with a procedural vote on Kavanaugh's nomina
to be pristine in order to function? No one knew, and Mr.
day, I answered questions under oath about every topic the Senat
Voss was not a man who welcomed questions. Early on, Anna had aske**

Figure 3. A poem produced by the Dada Poetry Generator

What the users don't see is the coded program enabling the deformation of the incorporated texts. The machine draws the content the users input into each data entry field. It engages in a process of random fragmentation pulling from each of the three input fields and reiterates these fragments to users in a poetic format. The code makes use of arrays and randomization functions in order to produce the recombination of lines into a poem (see Figure 4).

```

public function writePoem()
{
  //echo "The writePoem method is running<br>";
  for ($i=0; $i<$this->lines; $i++)
  {
    echo "<div align=\"center\"><span style=\"color:\".$rand_color.\">";
    $rand_index = array_rand($this->colors);
    //echo $rand_index."works";
    $rand_color = $this->colors[$rand_index];
    //echo $rand_color."works";
    $startpoint = rand(0,$this->stringlength);
    //echo $startpoint;

    $lineLength = rand(55,70);
    //echo $lineLength;
    $oneLine = substr($this->cleantext,$startpoint,$lineLength);
    //echo $oneLine;
    echo strip_tags($oneLine);
  }
}

```

Figure 4. A section of code from the Dada Poetry Generator

The Dada Poetry Generator is a machine inspired by the work of the Oulipo and Dada artists. Stephen Ramsay (2011) discusses the “combinatorial” possibilities of Oulipo works (p. 25) in his book *Reading Machines: Toward an Algorithmic Criticism*. Ramsay discusses the deformation of text as a means of creating new knowledge and meaning. The user-created poems in the Dada Poetry Generator are readable, although they sometimes may seem illogical. Ramsay suggests, “Deformation is a part of our permanent capacity for sense-making” (p. 48). Many Dada artists created works using found objects; a good example of this in poetry might be the cut-up poem, in which artists cut and paste different words traditionally out of newspapers in order to create a poem that may or may not be related to the original text. The Dada Poetry Generator uses a series of coded functions to replicate this process for the user. In so doing, it deforms the texts they provide in order to reassemble them in a way that has the potential to generate new meaning for the readers.

Dada artists were inspired by the current happenings in their daily lives, especially WW1. Similarly, the Dada Poetry Generator encourages users to engage with texts they are currently exploring in their daily lives. By examining the texts that are right in front of them, users might find new connections, insights, or pathways of thought that they may not have otherwise discovered. Ramsay discusses this idea in his chapter “Potential Readings.” While considering how transformations to text increases sense-making abilities he advises readers to “consider the patterns that emerge from various combinations of textual information” (Ramsay, 2011, p. 35). While he is speaking here of the information contained in a single text, it is easy to see how the combination of textual information grows when we combine multiple texts together as the Dada Poetry Generator does. In deforming the user’s texts, the Dada Poetry Generator provides a way for the user to misread the texts on purpose as a means of criticism and interpretation. The questions that appear directly be-

low the user-created poem aims to help readers gain insight from their new poems:

How would you describe the way the three texts fit together?

Which part of the poem makes the most sense?

Which part of the poem makes the least sense?

Did the texts fit together in way your anticipated? Why or why not?

Did the combination of the texts surprise you in anyway?

What are you now curious about? (Hill 2014)

These questions ask users to reflect on the creation and reading processes of their new poems as well as to think through the implications of meaning making within the combination of the texts. In deforming these texts, the users will most likely encounter the creation of nonsense. This nonsense is an opportunity for further exploration and meaning-making according to scholars like Marcel O’Gorman. O’Gorman (2006) suggests nonsense and play are integral to future study. He writes, “Various instances of nonsense...can short-circuit the rational world and act as short cuts between worlds of experience” (p. 81). Viewed another way, readers might find it useful to view the coded randomness of their texts as a process of what Victor Shklovsky calls “defamiliarization.” In his seminal essay “Art as Technique,” Shklovsky writes that the process of defamiliarization “makes the familiar seem strange” (p. 779). Indeed, the works created through the Dada Poetry Generator are works based in defamiliarization, that is to say, a work that is created artistically: “A work is created ‘artistically’ so that its perception is impeded and the greatest possible effect is produced through the slowness of the perception” (p. 783). The algorithms used to produce the new poems in the Dada Poetry Generator use code to produce artistic works, bringing pattern and randomness together to make the reader view familiar texts without “deautomatized perception” (p. 783), and allowing for “short cuts between worlds of experience” to create meaning.

Elza Adamowicz suggests one of the ways the meaning of verbal collages can be identified is through an identification with the “appropriated texts” (p. 16). If the collaged text is seen as sharing the identifying markers of the original three texts, it is logical to assume some of the identity can be unified back into the original text. The three texts a user chooses will relate to each other in different ways. They might all focus on the same topic(s); or they might cover completely different ones. The Dada Poetry Generator provides a way

to make a connection to various branches of the user's "world of experience" by visually placing them in close proximity to one another. Additionally, the new texts might highlight pieces of the originals that may have gone unnoticed, taking us, as O'Gorman suggests "across cultural and cognitive fields" (p. 81) in a way only nonsense can. By connecting texts from different areas of our lives, we make leaps from one subject to another and gain the opportunity to find common themes and patterns that are emerging in our daily lives and our society. This is the ultimate goal of the Dada Poetry Generator.

Discussion

In applying a dialect of pattern and randomness as an interpretative lens to e-poetry, we hope to offer readers and students of e-poetry a powerful tool to understand the complex nature of works created in the digital medium. We presented interpretations of our own two projects, "Wayfarer's Song" and the Dada Poetry Generator, to demonstrate that paying attention to the interplay of pattern and randomness can deepen our understanding of e-poetry and, by extension, digital works of literature and art.

In Case One, we examined the e-poem "Wayfarer's Song" with an eye to emerging patterns and instances of randomness. Pattern in "Wayfarer's Song" is at its most obvious in the structure of the poem: the Villanelle scheme. Randomness in "Wayfarer's Song" is at its most obvious on the level of code: the randomizing computer algorithm, modelled on the 'analog' algorithms used by Oulipo, determines the order of verses and thus, with each iteration, alters the possibilities of interpretation. But these instances of pattern and randomness only touch upon the surface layer of meaning. E-poems like "Wayfarer's Song" comprise not only all of the unwritten iterations of the poem - all the possible combinations, the *potential* versions of the poem - but it is also a .php file on a server, a link, a randomized algorithm, a nanoscopic etching on someone's hard drive, all of the flickering, effervescent images it produces on the screen, as well as the traces it leaves in the caches of browsers across the world. Meaning thus emerges from the interplay of random factors and patterned information across content, code, interface, and hardware. The pattern-randomness lens help make sense of poetry machines like "Wayfarer's Song:" poetry machines produce potential meanings, they are meta-work which comments on the nature of the digital medium and its complexity.

Case Two showcased the Dada Poetry Generator and demonstrated how readers can make meaning from textual material with no apparent syntactic pattern. The case showed how poems generated from algorithmic fragmentation, automation, and recombination have the power to ignite a new way

of exploring poetic works. Meaning and, by extension, pattern, emerge from readers' interpretation and the connections they make.

Together, the two cases illustrate that attending to the interplay of pattern and randomness in e-poetry can serve as an invaluable method of understanding the complex and multifaceted layers of meaning in digital texts. In future projects, we hope to apply the approach to other electronic works, including electronic art, different genres of electronic literature, as well as digital games. In the meantime, we hope that scholars and students in the fields of digital humanities, digital media, electronic literature, and digital art will find our interpretative lens useful in reading, composing, and analyzing electronic poems and other electronic artforms.

References

- Adamowicz, Elza. (1998). *Surrealist collage in text and image*. Cambridge, UK: Cambridge University Press.
- Emerson, Lori. (2014). *Reading, writing, interfaces. From the digital to the bookbound* [kindle edition]. Minneapolis, MN: University of Minnesota Press.
- Hayles, N. Katherine. (1999). *How we became posthuman: Virtual bodies in cybernetics, literature, and informatics*. Chicago, IL: University of Chicago Press.
- Hayles, N. Katherine. (2005). *My mother was a computer: Digital subjects and literary texts*. Chicago, IL: University of Chicago Press.
- Hayles, N. Katherine. (2008). *Electronic literature: New horizons for the literary*. Notre Dame, IN: University of Notre Dame Press.
- Hill, Amanda. (2014). Dada poetry generator [electronic poem]. Retrieved from <https://bit.ly/2HL9CO5>
- Kirschenbaum, Matthew K. (2012). *Mechanisms*. Cambridge, MA: MIT Press.
- Montfort, Nick. (2009). Taroko Gorge. In Stephanie Boluk, Leonardo Flores, Jacob Garbe, & Anastasia Salter (Eds.), *Electronic literature collection, volume III*. Cambridge, MA: Electronic Literature Organization. Retrieved from <http://collection.eliterature.org/3/works/taroko-gorge/taroko-gorge.html>
- O'Gorman, Marcel. (2006). *E-crit: Digital media, critical theory and the humanities*. Toronto, CA: University of Toronto.
- Okkema, Laura. (2016). Wayfarer's song [electronic poem]. Retrieved from <http://scriboergosum.net/test.php>
- Pequeño Glazier, Loss. (2008). *Digital poetics: The making of e-poetries*. Tuscaloosa, AL: University of Alabama Press.
- Queneau, Raymond. (1961). *Cent mille milliards de poèmes*. Paris, FR: Gallimard.
- Ramsay, Stephen. (2011). *Reading machines: Toward an algorithmic criticism*. Urbana, IL: University of Illinois Press.
- Ray, Robert. (1995). *The avant-garde finds Andy Hardy*. Cambridge, MA: Harvard University Press.
- Rettberg, Scott. (2009). Frequency. In Stephanie Boluk, Leonardo Flores, Jacob

- Garbe, & Anastasia Salter (Eds.), *Electronic literature collection, volume III*. Cambridge, MA: Electronic Literature Organization. Retrieved from http://collection.eliterature.org/3/works/frequency/frequency_info/frequency_wordlist.html
- Rettberg, Scott. (2014). Electronic literature. In Marie-Laure Ryan, Lori Emerson, & Benjamin J. Robertson (Eds.), *John Hopkins Guide to Digital Media*. Baltimore, MD: John Hopkins University Press.
- Shklovsky, Victor. (2007). Art as technique. In David Richter (Ed.), *The critical tradition* (pp. 775–784). Boston, MA: Bedford/St.Martin's.
- Strickland, Stephanie. (2009, February 13). Born digital. A poet in the forefront of the field explores what is—and is not—electronic literature [essay]. Poetry foundation. Retrieved from <https://www.poetryfoundation.org/articles/69224/born-digital>