



HAL
open science

A Literary Analysis of Poems Automatically Produced by Peter's Haiku Generator

Vinicius Carvalho Pereira, Cristiano Maciel

► **To cite this version:**

Vinicius Carvalho Pereira, Cristiano Maciel. A Literary Analysis of Poems Automatically Produced by Peter's Haiku Generator. 16th International Conference on Entertainment Computing (ICEC), Sep 2017, Tsukuba City, Japan. pp.24-31, 10.1007/978-3-319-66715-7_3 . hal-01771295

HAL Id: hal-01771295

<https://inria.hal.science/hal-01771295v1>

Submitted on 19 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Literary Analysis of Poems Automatically Produced by *Peter's Haiku Generator*

Vinicius Carvalho Pereira¹✉ and Cristiano Maciel¹

¹ Universidade Federal de Mato Grosso, Cuiabá MT, Brazil
{viniciuscarpe;crismac}@gmail.com

Abstract. The development of computer technologies gives rise to a revolution in all socio-cultural practices, including arts and literature. In terms of poetry, for example, readers have now access to new lyrical experiences either through emerging poetic genres or through so-far unimagined approaches to traditional ones. Lots of research has been done in the field of Computer Science on technical issues of computer-generated poetry, but little on the aesthetic impacts of having poems produced by machines. Therefore, we herein propose a multidisciplinary literary analysis of poems produced by the tool Peter's Haiku Generator, in order to discuss how software design decisions can affect literary aspects and the aesthetic effects of those poems.

Keywords: literature and new technologies · text generators · computer-based poetry · Peter's Haiku Generator.

1 Introduction

The development of computer technologies gives rise to a revolution in socio-cultural practices, including arts and literature. Along with a subversion of the literary system, in which the roles of authors and readers become blurry, in a prosumer paradigm (Kim et al [11]), the materiality of poetic works is also altered, since the digital and the printed media develop increasingly distinct aesthetic resources. Analyzing the research area of entertainment computing in industry and academy, Nakatsu and Rauterberg [16] highlight “art and entertainment” as one of the seven active working groups in TC14 (Entertainment Computing) by IFIP, where an interdisciplinary approach is highly desired.

The printing press revolutionized the literary art of the Modern Age, providing it with agile circulation and effective reproducibility. The emergence of the novel as a longer literary genre, due to the invention press, is one of the most noticeable effects of how technology influences much more than the contents of a literary text. McLuhan [14] agrees with that in his most famous quoting: “the medium is the message”.

Likewise, the new information and communication technologies radically affected the symbolic practices of our increasingly digital society, which is immersed in what Lévi [12] calls cyberculture. Like any other field, such paradigm affects also literature, which causes the emergence of new genres, some of which compose an exclusively virtual poetics. That is the case of computer-generated poetry (grouped under the

wider term "Generative Literature"), which consists of poems produced by automatic recombination of elements of different granularity, following algorithms that consider natural language processing. To Zhang and Lapata [25], automated poetry generation has been a popular research topic over past decades, but we herein discuss how software design decisions can affect literary aspects and the aesthetic effects of those poems. To do so, we chose the tool *Peter's Haiku Generator*, among others available online, because the genres of poem it produces (haiku and tanka), in comparison to most poetic forms in the West, are very short and and rely on sometimes fragmented visual images, which are features easier to meet in passable quality by computer-generated poems. Besides, different from most other online haiku generators, this tool produces haiku in English, instead of Japanese, which makes it easier for a broader audience to follow the discussions herein presented.

This paper is structured as follows: after the introduction, we present in section 2 a brief theoretical discussion about generative literature and a review of related works about haiku generators. Section 3 presents an overview of Peter's Haiku Generator and a close reading of two of its poems. Section 4 contains the final remarks of the paper and is followed by the references section.

2 Algorithmic Literature: Computer-Generated Haiku

In the area of Literature Studies, research on computer-generated literature is still scarce, since the idea of machines producing poems deconstructs some of the most traditional understandings of poetry as an emotional expression that comes from inspiration. Therefore, before we move to a literary analysis of the poems generated by *Peter's Haiku Generator*, it is necessary to specify what notions of language and literature are at stake when discussing computer-generated poems.

An analysis of literary texts generated by computers must be grounded in the assumption that language itself is an algorithmic phenomenon, as proposed in Structural Linguistics [21], Generative Grammar [3] and Computational Linguistics [15]. According to those approaches, every language is formed by a grammar, that is, a set of associative and restrictive principles that command the selection and ordering of lexical items, similarly to the data a computer processes following combination rules [13]. Furthermore, as in the digital information cryptography into binary systems, every natural language also operates by means of opposite relations, according Saussure [21].

As to the definitions of literature herein adopted, Barthes [2] considered that literature is a cheating within language rules, thus deconstructing traditional forms of perception and representation. To do so, literature brings linguistic systems to limit situations, forcing new word associations that are possible according to language rules, but not commonly used in daily communication. Such understanding of literature supports the idea that a machine, equipped with the structures and the lexicon of a language, can form verses by combinatory rules, since it will operate within the limits imposed by language itself and coded by the software programmer. And, if the verses formed seem sometimes unfit, or even nonsensical, this does not make their aesthetic value null, since these awkward formations can be read as the cheatings within language argued for by

Barthes. Considering that estrangement is a formalist category associated with literariness [10], an awkward verse produced by the machine could be read not only as an accidental malformation, but as a token of what makes a text literary: the capacity to make people think about language, denaturalizing the relationship between things and words. Besides, as computer-generated poems frequently lead to somewhat disjointed verses, it is important to highlight that Eco [5] stated that semiosis comes not only from respecting, but also from disrespecting established codes, so there is meaning even in noise. This is why Manurung *et al.* stated that “poetry generation is different from traditional informative generation due to poetry’s unity, which essentially means the satisfying of interdependent constraints on semantics, syntax and lexis.” [14].

Accepting the hypothesis that texts randomly produced by machines can be poetic, Generative Literature must be read and analyzed considering the characteristics of this emerging genre, imposed by its enunciation format: instead of a poetic product whose full message is a logical product, coherent in its parts, as in more traditional literary texts, fragmentation is inherent to this virtual poetic.

Another problem that this genre poses to the traditional literary system has to do with authorship. Far beyond the death of the author professed by Barthes [1], who defended people should read texts ignoring authorial intention, Generative Literature goes further: it dissolves notions of authorship as understood in the area of Literature Studies. In summation, who is the author of a software-generated poem? The reader that unchains the poem’s random composition by simply clicking a button? The software engineer who also wrote a text - the system code -, which allows the functioning and artistic operation of the machine? Or the author of the original verses recombined by the software, in the case of machines that work with this feedstock? Among other software programs of this kind, the group ALAMO (*Atelier de Littérature Assistée par les Mathématiques et les Ordinateurs*) developed the software *Alexandrins au greffoir* [19], which forms sonnets from the random recombination of halves of alexandrine verses by famous French poets, such as Nerval, Valéry and Baudelaire.

The problem is that none of these agents knows the final poetic product before the automatic recombination happens. If authorship is decentralized among agents who do not know the final product of their work, it is because software-generated poems make language itself speak, minimizing the effects of authorial intention. Thus, what is poetic is the process through which the poems are generated, rather than each generated text itself, since each of them is nothing but one among countless combinations the machine could generate. Furthermore, they vanish forever by means of a single click if they are not saved in a separate file, in case some reader wants to perpetuate a poetic moment.

Whereas the romantic tradition and the 19th century criticism valued the image of the author as that of a person touched by inspiration, away from the common people and master of the meanings of his text, Generative Literature deconstructs all these parameters, leading us to re-think the act of writing itself. Compared to a text engineer, the poet is no longer someone who works guided by the muses’ whispers, but an individual that operates linguistic (and computer-based) algorithms, aiming at an aesthetic effect. Surprised with a machine’s random combinations, the system developer who created the algorithms becomes a common reader of the poems. And other readers who

click the refresh button generate new poems, so they play a physical part in the poetic production, thus contributing as co-authors. But who is the writer and who is the reader then? Maybe, such aesthetic claims for a new paradigm: that of the wreeder [8].

Among different poetic genres that can be produced by automated text generators, haikus tend to have more effective aesthetic results in comparison to most poetic forms in the West, since haikus are very short and rely on visual images, which are features easier to meet in passable quality by computer-generated poems. According to Obara, Tosa and Minoh [18], haiku is a “Japanese classical poem style with minimal length of 5-7-5 characters” and “it is a story that generates context – the shortest story in the world”. In English, as in any other Indo-European languages, haikus are composed of three verses, with 5-7-5 syllables respectively.

Tools to produce software-generated poems are not new. The first of them, created by Theo Lutz in 1959 [13], did not allow any input from users. More recent tools, on the other hand, allow users to interact with the system, such as *Hitch Haiku* (2007 - 25) and its developments [22][23]; others use Word Associations Norms, such as *Gaiku* [17]. There are also systems that combine traditional poetry culture and mobile messaging technology, such as *Poetry Mix-up* [6] and others which use word occurrences in the WWW and grammar templates, as proposed by [20].

Recently, Hreskova and Machová [24] carried out research comparing two different approaches to the automated generation of haiku poetry. Their first approach consists of evolutionary algorithms and human as a fitness function in the evolution. The second approach consists in generating poems based on haiku models that were extracted from haiku database. The goal is to create poems, considered by humans as understandable and with aesthetic value. In this research, when comparing poems generated by both systems, generating haiku poetry with poem models created better poems both in terms of form and content. According to the authors, one of the reasons is that it is easier to combine words than whole verses.

3 *Peter's Haiku Generator and Two of its Poems*

Peter's Haiku Generator is a system programmed in Flash that generates haikus and tankas automatically whenever a user clicks the refresh button. The system does so through a series of algorithms that rule the combination of words registered in its database. These words are categorized according to their parts of speech. The initial step taken by the software, after the user's click, is the random selection of a syntactic structure for each verse (there are 3 in a haiku, or 5 in a tanka). Although the software initially works with a predefined set of basic sentence patterns, those can also be enriched by the user, as long as he or she has some knowledge of Linguistics and Flash.

In linguistic terms, Howard opted for a generative model, where a structure called "simple sentence" can be divided into two substructures, called noun phrase and verb phrase, also breakable into even smaller substructures, and so on, until minimal non-breakable structures. Table 1 shows the basic syntactic structures and the respective substructures that can be employed by *Peter's Haiku Generator*. Brackets indicate

that a certain substructure is optional, depending on the number of syllables of the selected words, so as to meet the requirements of 5-7-5 syllables per verse, respectively:

Structures	Substructures
<sentence>	<clause> [<linking word> <clause>]
<clause>	[<gerund utterance>] <clause>
<clause>	1. <noun phrase> [<adverb phrase>] <intransitive verb> [<preposition> <noun phrase> 2. <noun phrase> <intransitive verb> [<preposition> <noun phrase>] [<adverb phrase>] 3. [<adverb phrase>] <noun phrase> <intransitive verb> [<preposition> <noun phrase>] 4. <noun phrase> <transitive verb phrase> <noun phrase>
<noun phrase>	1. <definite article> [<adjective phrase>] <common noun> 2. [<adjective phrase>] <noun> 3. <indefinite article> [<adjective phrase>] <common singular noun>
<gerund utterance>	1. [<adverb phrase>] <gerund verb>
<adverb phrase>	1. [<adverb>] <adverb>
<transitive verb phrase>	1. [<adverb phrase>] <transitive verb>
<adjective phrase>	1. <adjective> [" , " <adjective>]
<adjective>	1. <adjective> [<adjective>] 2. [<adverb>] <adjective>

Table 1. parsing rules processed by *Peter's Haiku Generator* [9]

After the random selection of the underlying syntactic structure, the system randomly selects words in its database to fill the previously defined syntactic substructures, which obey the standard English grammar. Furthermore, to warrant isotopies [7], i.e., coherence produced by words with meanings from similar areas, the words in the database are tagged according to the semantic field to which they belong. Before clicking the refresh button, the user can choose the semantic field of the poem to be generated; he or she can also add new words to the database, in case he or she has some knowledge about programming in Flash. Finally, an algorithm related to the metrics of the verses is triggered: only verses with seven and five syllables are formed according to the traditional versification of haikus and tankas.

Such idiosyncrasies of the machine's "writing" process, if compared to the way how poems are traditionally produced also demand particular methods for literary analysis. For example, it is not possible to exhaustively read the whole poetic production generated by the software, as it is frequently done in researches on human poets and their writings. The only methodological option, so to speak, is to randomly generate some poems and make a *close reading* of them, which means disregarding elements like authorial intention, literary period, social context and other features frequently considered in literary analyses.

For the sake of the literary analysis herein proposed, we chose the semantic field noir in the interface and clicked the refresh button twelve times before we reached the haiku below transcribed:

Yellow winds waste streets.
Fear turns however heat turns.
Tepid heaven burns. [9]

The choice for that specific semantic field justifies the presence of words associated to the notions of destruction and suffering, such as *waste*, *fear* and *burns*. The same meaning is conveyed in the phonetic layer of the verses by the assonance (vowel repetition) of the central vowel followed by the voiced alveolar trill, in *fear*, *turns*, *however* and *burns*. In those words, the sound resonates in the mouth cavity with little movement of the lips and tongue; they are inert as the desolate landscape described in the haiku. Also regarding stylistic phonetics, the alliteration of the semivowel /w/ (produced by the gliding movement of the lips) suggests the waves of warm wind that ravage the images described in the poem.

A poem *coldly* written by a machine, and not by an inspired author, paradoxically describes a warm scene, both in the phonological and in the lexical layers, as seen in the phrase "*tepid heaven burns*". That paradox is analogue to the one that connects programming and random in the construction of a haiku by a machine. Another paradox is produced by the synesthesia in the phrase "yellow winds", which describes winds that sweep the images of the poem just like the electric currents that erase each new haiku when the user clicks the refresh button.

But how can one make a literary interpretation of software generated by a machine? If each haiku is only the fortuitous product of chance, is it pertinent to search for different layers of meaning in their reading? Eco [4] defends a literary analysis based on the work's intention, rather than the author's intention or the reader's intention. According to that paradigm, every literary text is provided with an immanent intention and significance, which the analysis herein carried out tries to retrace.

4 Final Remarks

If Remington type-writers once revolutionized the way people write, today poetry generators revolutionize the way to understand writing itself: in generative literature, who writes? Who reads? What is the work of art: the software or the poem? In this sequence of questions, the definitions of literature and poetry are at stake, especially because the concepts of reader and writer do not really apply to generative literature.

Poems automatically produced by a machine require specific analytical procedures, which must understand poetry as a product of systems (both linguistic and computational ones) rather than inspiration. Therefore, we have opted for a close reading approach in this paper.

A new aesthetics paradigm is brought forth by generative literature, which requires new understandings of literature. In that context, some of the haikus produced by Peter's Haiku Generator can be read as metapoems, whose noise from awkward images is not a communication break. Rather than that, it is meaning formed by random word association, which should be read as the work's intention, and not as the author's.

References

1. Barthes, R.. A morte do autor. In: Barthes, R. *O rumor da língua*. São Paulo: Martins Fontes, 2004.

2. Barthes, R.. *Aula*. São Paulo: Cultrix, 1994.
3. Chomsky, N.. *Language and Mind*. New York: Harcourt Brace & World, 1986.
4. Eco, Umberto. *Interpretação e superinterpretação*. São Paulo: Martins Fontes, 2005.
5. Eco, Umberto. *The open Work*. Harvard: University Press, 1989.
6. Fernando, Owen Noel Newton, et al. "Poetry mix-up: a poetry generating system for cultural communication." *Proceedings of the International Conference on Advances in Computer Entertainment Technology*. ACM, 2009.
7. Greimas, A.J.; Courtès, J.. *Sémiotique; dictionnaire raisonné de la théorie du langage*. Paris: Hachette, 1993.
8. Hayles, N.K. *Electronic Literature: What is it?* V.1.0. Januay, 2007. <https://eliterature.org/pad/elp.html>
9. Howard, P.. *Peter's Haiku Generator*. Available at <<http://www.peterhoward.org/flash/haikugen.html>>.
10. Jakobson, R. *Linguística e comunicação*. São Paulo: Cultrix, 1995.
11. Kim, W., Jeong, O. R., Lee, S. W. (2010). On social Web sites. *Information systems*, 35(2), 215-236.
12. Lévy, P. *Cibercultura*. São Paulo: Editora 34, 2009.
13. Lutz, T.: Stochastische text. *Augenblick* 4(1), 3-9 (1959)
14. McLuhan, M.. *Os meios de comunicação como extensões do homem*. São Paulo: Cultrix, 2000.
15. Mitkov, R. (2005). *The Oxford handbook of computational linguistics*. Oxford University Press.
16. Nakatsu, Ryohei, and Matthias Rauterberg. "Entertainment computing: inaugural editorial." *Entertainment computing* 1.1 (2009): 1-7.
17. Netzer, Yael, et al. "Gaiku: Generating haiku with word associations norms." *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*. Association for Computational Linguistics, 2009.
18. Obara, Hideto, Naoko Tosa, and Michihiko Minoh. "Hitch haiku: an interactive generation system of haiku." *Proceedings of the international conference on Advances in computer entertainment technology*. ACM, 2007.
19. Pereira, V. C., Maciel, C. (2013, October). The user's and the designer's role and the aesthetic experience of generative literature. In *Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems* (pp. 188-197). Brazilian Computer Society.
20. Rafal, Rzepka, and Araki Kenji. "Haiku generator that reads blogs and illustrates them with sounds and images." *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*. IJCAI, 2015.
21. Saussure, F.. *Cours de linguistique générale*. Lausanne : Payot, 1955.
22. Tosa, Naoko, Hideto Obara, and Michihiko Minoh. "Hitch haiku: An interactive supporting system for composing haiku poem." *International Conference on Entertainment Computing*. Springer Berlin Heidelberg, 2008.
23. Wu, Xiaofeng, Naoko Tosa, and Ryohei Nakatsu. "New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system." *International Conference on Entertainment Computing*. Springer Berlin Heidelberg, 2009.
24. Hruskova, M.; Machova, K. Interactive Evolution and Poem Models in Haiku Poetry Generation. IN Bieliková, M.; SRBA, I.. Použitie transformačnej regresnej techniky pre določovanie v údajoch. In *11TH Workshop on Intelligent and Knowledge Oriented Technologies*, pp.103-108, 2016.
25. Zhang, Xingxing, and Mirella Lapata. "Chinese Poetry Generation with Recurrent Neural Networks." *EMNLP*. 2014.