Computational Poetry is Lost Poetry

Max Kreminski
mkreminski@midjourney.com
Midjourney
San Francisco, California, USA

ABSTRACT

The genre of "found poetry" encompasses passages of text that were first framed as poetry by someone other than the original writer. But for something to be found, it must first be lost. In an attempt to shed light on the role of computational intelligence in the human creative ecosystem, I argue that many computational generators of poetry function essentially as *poetry losers*: machines whose central purpose is to arrange units of language, without fully understanding them, in combinations that can later be found to be poetry. This implies a paradox for computational poetry: a poetry machine that too completely understands the poetic effects of its output deprives human readers of the chance to *find* poetry where the machine did not, fundamentally altering both the reader's poetic experience and the machine's utility. I briefly explore the implications of this view, taking computational poetry as a microcosm of "intelligent" machines in creative contexts generally, and discuss what it means to construct an effective poetry loser.

CCS CONCEPTS

• **Applied computing** → **Arts and humanities**; • **Human-centered computing** → Human computer interaction (HCI).

KEYWORDS

computational creativity, creativity support tools, generative AI, poetry

ACM Reference Format:
Max Kreminski. 2024. Computational Poetry is Lost Poetry. In *Halfway to the Future (HTTF '24), October 21–23, 2024, Santa Cruz, CA, USA*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3686169.3686179

1 INTRODUCTION

Belief in computation's creative potential has existed for almost as long as computers themselves [6], but the recent growth of mainstream interest in computational creativity has largely been driven by a relatively narrow class of technologies: generative machine learning models that reify the statistical shadows cast by human creative output, predominantly via distillation of patterns from large corpuses of human-crafted creative works [11, 28]. These technologies have been used to model creative forms as diverse as visual art [3], music [17], game levels [27], stories [33], and poetry [12]; across these forms, large-scale data-driven generative models tend to exhibit similar strengths (e.g., flexible adaptation to a wide range of creative domains [32]) and weaknesses (e.g., unoriginality [1]).

In this essay, I wish to call attention to a recurring characteristic of computationally creative systems that has stubbornly persisted through the shift toward large-scale data-driven approaches: the overall failure of machines to fully model the likely aesthetic effects of their output on human observers. As argued by Winograd and Flores [31] in response to earlier (rules-based) attempts at artificial intelligence, all hitherto existing machines have lacked a number of perceptual faculties that humans generally possess, limiting their ability to extrapolate the consequences of speech acts and thus their ability to participate fully in human language. Modern large-scale generative ML systems seem to reconstruct a much greater proportion of the semantic associations that characterize human "common sense" [24], but still frequently fail to (for instance) extrapolate implied consequences of imaginary situations that humans immediately perceive [29]. These machines can thus still be criticized along similar lines—including in the context of creativity, where the ability to predict an artifact's likely aesthetic effects on a human audience seems critical to success [16].

In highlighting this failure, however, I do not mean to suggest that machines cannot be usefully creative. Instead, I argue that machines are often usefully creative *because* they fail to see things completely as humans do: their oversights and inabilities lead them to mix human-like with non-human-like creative decisions in unanticipated ways, and thereby to supply human creators with ideas that they otherwise never would have considered. Somewhat counterintuitively, then, I suggest that a dogged pursuit of perfect overlap between human and machine understanding of aesthetic domains may in fact inhibit the usefulness of machines as generators of unexpected inputs to the human creative ecosystem. A hypothetical machine that takes into account all the same considerations as a human artist would produce outputs very similar to what humans can produce unassisted already. Instead, the most valuable input generators are those computational systems that are (deliberately or accidentally) left unable to anticipate some portion of their output's aesthetic effects.

As a microcosm of this dynamic, I turn now to computational poetry, and its prototypical status (in my view) as *lost poetry*.

2 LOST POETRY

Found poetry (Fig. 1) is poetry produced by an agent or process that did not intend to generate poetry. It consists of language generated toward some other end, creatively reframed as poetry by a later recipient [9]. *Lost poetry*, then, is language with poetic potential, not originally intended by its creator for poetic interpretation, that may or may not later be *found to be* poetry.

HTTF '24, October 21–23, 2024, Santa Cruz, CA, USA

ACM ISBN 979-8-4007-1042-1/24/10
https://doi.org/10.1145/3686169.3686179

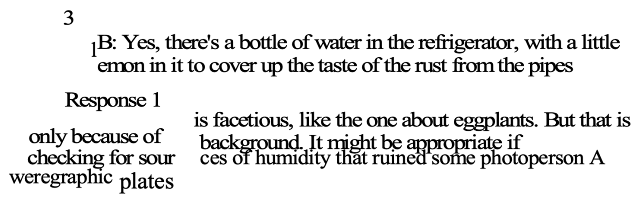

Figure 1: An example of lost-and-found poetry from a digitally mangled PDF of *Understanding Computers and Cognition* [31]. Neither the original authors of the text nor the digitization process intended to arrange these letters in precisely this way, but the overall effect of passages like "only because of / checking for sour / weregraphic plates" remains quite vivid and poetic.

This term, however, also suggests a range of largely but incompletely overlapping interpretations. Lost poetry could alternatively be viewed as "poetry that is lost on its creator", or even "the poetry produced by an agent or process that is [semantically] lost". The computational poet Allison Parrish frames poetry machines as "semantic space probes", robots that we send ahead of us to report back on (semantic) environments that we lack the capability or wherewithal to traverse ourselves [22]; in this sense, lost poetry can be seen as poetry that has been lost in (semantic) space.

The treatment of computational poetry as lost poetry may not at first be obvious: if a machine is constructed with the goal of generating poetry, is the poetry that it generates truly lost? I argue that it is, for the simple fact that the machine does not comprehend the full poetic effects of its output. This is closely related to Parrish's contention that language models can *only* generate poetry [23]: much as Winograd and Flores argued of older attempts at AI, language models cannot commit speech acts through the language they generate, so this language is assigned the fallback status of poetry. Indeed, Parrish asserts, the language model's lack of intentionality ensures that it does not even function *as a poet*: though the machine can arrange language in ways that humans sometimes find to be poetic, it is not ultimately responsible for the framing of its own outputs as poems.

The most enduring products of poetry machines are often clear instances of lost poetry: something important about these poems was *lost on* (i.e., went undetected by) the computational processes that generated them. The "mis-spun tales" (Fig. 2) produced by early story generators such as Tale-Spin [18] stand out as these systems' most memorable outputs [30]; the standout outputs of the erasure poetry generator Blackout [15] also tend to be examples of lost poetry (Fig. 3), in which the machine's obliviousness to paragraph-level juxtaposition has led it to strangely fortuitous sequencings of technically independent paragraph-level choices.

These remarkable outputs stand out not just against the space of existing human poetry, but also against the *expressive range* [26] defined by the poetry machine's typical output. As in procgen poetics [13], viewing a single output against the backdrop of what the machine was presumably "meant" to produce can call to attention something unanticipated about the output in question. Discovering a particularly striking assembly of language in the output of a poetry machine is synonymous with finding an unexpected overlap between one's own poetic perceptions and the perceptions of an alien system; from a framing perspective [4, 7], this overlap acts as an indication of agreement between multiple independent observers that something poetically worthwhile is here.

> Henry Ant was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned.

Figure 2: A famous "mis-spun tale" produced by the Tale-Spin story generator [18]. The generator models "gravity" as an agent just like the other characters in the story; agents in general can be called up to perform any action; and the semantic associations of agent *names* are completely opaque to the generator, allowing it to produce the otherwise-improbable but strikingly poetic bigram "Gravity drowned."

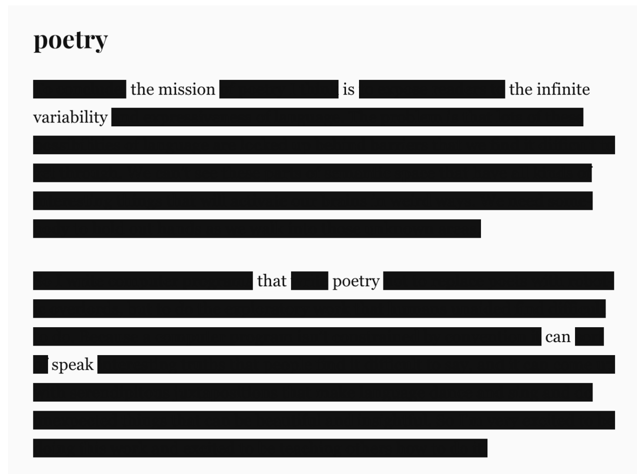

Figure 3: An exemplary output of the erasure poetry generator Blackout [15], taken from the generator's webpage. Blackout performs erasure of each paragraph from a source text independently, without taking into account the other paragraphs at all; thus the apparent joining of these two erased paragraphs into a single, unitary work of poetry represents an instance of poetic effect being lost on the machine.

For all of these reasons, it seems clear to me that **computational poetry is lost poetry**. I therefore characterize poetry machines as producers of lost poetry: in other words, as *poetry losers*.[1] One additional aesthetic effect of the poetry produced by losers is perhaps best captured by the following excerpt from Brian Eno's *A Year With Swollen Appendices* [8]:

> Whatever you now find weird, ugly, uncomfortable and nasty about a new medium will surely become its signature. CD distortion, the jitteriness of digital video, the crap sound of 8-bit—all of these will be cherished and emulated as soon as they can be avoided. It's the sound of failure: so much modern art is the

[1] I myself am also a poetry loser. Maybe you are too?

> sound of things going out of control, of a medium pushing to its limits and breaking apart. The distorted guitar sound is the sound of something too loud for the medium supposed to carry it. The blues singer with the cracked voice is the sound of an emotional cry too powerful for the throat that releases it. The excitement of grainy film, of bleached-out black and white, is the excitement of witnessing events too momentous for the medium assigned to record them.

The best outputs of semantic space probes are given further poetic depth by the poignancy of their success-through-failure. "Loser" is an especially apt appellation for poetry machines because all existing poetry machines are characterized partly by their own inadequacy at the task of understanding their own poetry.

3 WHY TO GET LOST

The most important function of lost poetry, I argue, is to lead human poets to places that they would not otherwise explore. Artists of many stripes, poets included, often express a desire—even a need—to somehow escape from the realm of the expected. Human adaptation to repetition results in arrangements of words that once profoundly expressed important sentiments gradually losing their strength and becoming cliché. Consequently, poets must always be on the lookout for new ways of arranging language—even just to powerfully express the same old sentiments, let alone to give voice to new ones. And yet clichés persist precisely because human poets find them ready-to-hand: it is simply more difficult to find new ways of creating meaning through language than it is to reuse well-tested constructions from poetry past.

Thus, poets try to lose themselves. From the Oulipan use of constraints to taboo vast swaths of likely word-arrangements [20] to the Situationist translation of literal, physical lostness to artistic lostness via the dérive [19], the history of poetry is in part a history of finding new ways to become semantically lost.

Our urgency to get lost seems to imply to me that we seek not just to lose ourselves but also to *lose our pursuers*: in particular the specter of perceived sameness or *perceptual collapse*, which has been argued in the past to haunt all generated media [14]. Purely human art, too, struggles with sameness: it is difficult to escape from aesthetic trends and fashion cycles, not least because human creativity is at least substantially recombinatory. Surrounded by examples of how things are currently done and buffeted by the stylistic whims of editors and audiences, escape from the oblivion of blending indistinguishably into the aesthetic *moment* may feel all but impossible.

The successful poetry machine's unfamiliarity—with poetic convention, with intertextual reference, with the full aesthetic effects of particular word arrangements, and so on—allows it to clip outside the bounds of present human expectation. In its innate lostness, the poetry loser tries things that usually don't work, or that wouldn't seem likely to work on the surface from a human perspective; in the process, it does not always succeed, but it sometimes discovers arrangements of language that are startlingly vivid, poignant, or apt. Blithely bounding through the semantic minefield, ignorant of the potential consequences of poetic failure, the machine nevertheless attempts to chart a path to its goal. When it does not blow up along the way, it may reveal to us a new way of traversing the fraught semantic space, to safety—however temporary—from the creeping bland sameness of poetic cliché.

4 CONCLUSION

Investigation of computational poetry reveals that, in creativity, machines can win by losing. Making them better at finding things, to align them with human finding abilities, runs the risk of making them worse at losing—and thus less able to win.[2]

This suggests that the design of novel computationally creative systems could be guided in part by a deliberate choice of what to make *invisible* to the machine. By selectively limiting the machine's capacity to take certain facets of human aesthetic perception into account, we can produce different kinds of losers that can help to break us out of familiar patterns toward new techniques of expressive communication. Furthermore, this strategy seems likely to generalize across a wide range of creative domains: many visual artists, for instance, are just as infatuated as poets with the glitchy or off-kilter outputs of computationally creative systems [2, 3]. Consideration of lostness as an explicit *goal* of human artists may even drive the design of new creativity support tools [5, 10, 25] that do not incorporate generative AI. The future looks bright for losers.

ACKNOWLEDGMENTS

Thanks to all the poetry losers who helped me formulate the ideas found in this essay—especially Barrett R. Anderson, Phoebe J. Wang, and Melanie Dickinson.

REFERENCES

[1] Barrett R Anderson, Jash Hemant Shah, and Max Kreminski. 2024. Homogenization Effects of Large Language Models on Human Creative Ideation. In *C&C '24: Proceedings of the 16th Conference on Creativity and Cognition.*

[2] Baptiste Caramiaux and Sarah Fdili Alaoui. 2022. "Explorers of Unknown Planets": Practices and Politics of Artificial Intelligence in Visual Arts. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022).

[3] Minsuk Chang, Stefania Druga, Alexander J Fiannaca, Pedro Vergani, Chinmay Kulkarni, Carrie J Cai, and Michael Terry. 2023. The prompt artists. In *Proceedings of the 15th Conference on Creativity and Cognition.* 75–87.

[4] John William Charnley, Alison Pease, and Simon Colton. 2012. On the Notion of Framing in Computational Creativity.. In *International Conference on Computational Creativity.* 77–81.

[5] John Joon Young Chung, Shiqing He, and Eytan Adar. 2021. The intersection of users, roles, interactions, and technologies in creativity support tools. In *Designing Interactive Systems Conference 2021.* 1817–1833.

[6] Simon Colton and Geraint A Wiggins. 2012. Computational creativity: The final frontier?. In *ECAI*, Vol. 12. Montpelier, 21–26.

[7] Michael Cook, Simon Colton, Alison Pease, and Maria Teresa Llano. 2019. Framing in Computational Creativity – A Survey and Taxonomy. In *International Conference on Computational Creativity.* 156–163.

[8] Brian Eno. 1996. *A Year with Swollen Appendices: Brian Eno's Diary.* Faber & Faber.

[9] Laurence Foss. 1972. Poetry is Where You Find It: "Found Poetry". *Journal of Popular Culture* 5, 4 (1972), 821–830.

[10] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the landscape of creativity support tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.*

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Vol. 33. 6840–6851.

[12] Zhiyuan Hu, Chumin Liu, Yue Feng, Anh Tuan Luu, and Bryan Hooi. 2024. PoetryDiffusion: Towards Joint Semantic and Metrical Manipulation in Poetry Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 18279–18288.

[2] For instance, finetuning language models to align them with human preferences seems to reduce their capacity for originality. [21]

[13] Isaac Karth. 2019. Preliminary poetics of procedural generation in games. *Transactions of the Digital Games Research Association* 4, 3 (2019).
[14] Max Kreminski. 2023. "Generator's Haunted": A Brief, Spooky Account of Hauntological Effects in the Player Experience of Procedural Generation. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*.
[15] Max Kreminski, Isaac Karth, and Noah Wardrip-Fruin. 2019. Generators that read. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*.
[16] Max Kreminski and Chris Martens. 2022. Unmet creativity support needs in computationally supported creative writing. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*. 74–82.
[17] Ryan Louie, Jesse Engel, and Cheng-Zhi Anna Huang. 2022. Expressive communication: evaluating developments in generative models and steering interfaces for music creation. In *27th International Conference on Intelligent User Interfaces*. 405–417.
[18] James R Meehan. 1977. TALE-SPIN, An Interactive Program that Writes Stories. In *IJCAI*, Vol. 77. 91–98.
[19] Tyrus Miller. 2012. Lettrism and Situationism. In *The Routledge Companion to Experimental Literature*. Routledge, 101–114.
[20] Nick Montfort and Noah Wardrip-Fruin. 2003. Six Selections by the Oulipo. In *The New Media Reader*, Noah Wardrip-Fruin and Nick Montfort (Eds.). MIT Press, 147–148.
[21] Vishakh Padmakumar and He He. 2024. Does Writing with Language Models Reduce Content Diversity?. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=Feiz5HtCD0
[22] Allison Parrish. 2015. Exploring (Semantic) Space With (Literal) Robots. http://opentranscripts.org/transcript/semantic-space-literal-robots.
[23] Allison Parrish. 2021. Language models can only write poetry. https://posts.decontextualize.com/language-models-poetry.
[24] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 2463–2473. https://doi.org/10.18653/v1/D19-1250
[25] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.
[26] Gillian Smith and Jim Whitehead. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*.
[27] Graham Todd, Sam Earle, Muhammad Umair Nasir, Michael Cerny Green, and Julian Togelius. 2023. Level generation through large language models. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*.
[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30.
[29] Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. 2024. Can Language Models Serve as Text-Based World Simulators? *arXiv preprint arXiv:2406.06485* (2024).
[30] Noah Wardrip-Fruin. 2013. Reading digital literature: Surface, data, interaction, and expressive processing. In *A Companion to Digital Literary Studies*, Ray Siemens and Susan Schreibman (Eds.). Blackwell, 161–182.
[31] Terry Winograd and Fernando Flores. 1986. *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley.
[32] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7378–7385.
[33] Anbang Ye, Christopher Zhang Cui, Taiwei Shi, and Mark Riedl. 2023. Neural Story Planning. In *The AAAI-23 Workshop on Creative AI Across Modalities*. https://openreview.net/forum?id=cLBEKlu5WZK