



# Lotus Chorus Workshop: Designing for Cognitive Overload

**Kyle Booten**

[kylebooten@uconn.edu](mailto:kylebooten@uconn.edu)

University of Connecticut, Storrs, USA

**DOI** [10.34626/xcoax.2023.11th.185](https://doi.org/10.34626/xcoax.2023.11th.185)

This paper describes Lotus Chorus Workshop, a text-based interface that assists poets by simulating — albeit abstractly and sometimes absurdly — a creative writing workshop. Unlike typical creativity-support tools, Lotus Chorus Workshop is polyvocal: its various characters respond to a user’s poem with different (and possibly even contradictory) suggestions. An example is provided of poetry written in response to the system’s critical remarks and suggestions. While mainstream digital tools for helping writers tend to try to streamline the user’s writing and thinking, this paper considers the literary affordances of software that pushes the user’s mind in too many directions at the same time.

**Keywords:** Poetry, Creative Writing Workshop, Creativity Support Tools, Natural Language Processing, Cognitive Overload.

## 1. What Is a Workshop?

Lotus Chorus Workshop is a creativity-support tool that simulates a creative writing workshop. Since the creative writing workshop, while nearly ubiquitous in the United States, is not universally so, I should begin by describing this pedagogical model.

The creative writing workshop bears a resemblance to the art school “crit.” A handful or two of students sit around a table. One of them is “being workshopped”; this student’s writing — today, a poem — is the topic of discussion, and classmates and the instructor alike have arrived with a marked-up copy of their work.

If this is an undergraduate workshop, these students are in attendance for a variety of reasons. A few of them think of themselves as “serious” writers, a few simply want to follow up on an interest from secondary school, and a few more are hoping for a low-stress interlude between gruelling problem sets for their math and science classes. If this is a postgraduate program, the graduate students all think of themselves as more or less “serious,” but it is still a mixed bag. Each workshop participant has different tastes and favorite authors, and each has at least a somewhat distinct background and personality.

The “workshopping” itself is straightforward enough: students and the instructor express their opinions about what they like about the poem, what they do not, what confuses them, what confuses them pleasantly, and all sorts of other aesthetic matters. But it is here in the nitty-gritty of analyzing and judging a poem that the irreducible heterogeneity of the workshop — that is to say, the irreducible heterogeneity of any group of interlocutors — becomes apparent. Only rarely is there anything like a consensus: *You should have ended the poem here, scrap the last line*, and everyone else nods, or *Frankly, I found this word choice to be offensive*, and everyone else nods. Typically, one commenter thinks the poem’s last line is spot-on, while another complains that it is too heavy-handed. One thinks a line would be perfect if only “mauve” were changed to something less dowdy, while another would prefer that this entire stanza be mulched. The instructor also offers an opinion but typically does not adjudicate.

The writing workshop is the dominant pedagogical form, the sine qua non, of contemporary creative writing instruction in the United States, and rare is the department of English literature that does not offer at least one. That this mode of instruction has become so popular, suggests that what it offers is valuable. And what it offers, primarily, is differences of opinion about what a poem means, what feelings it provokes, and what steps should be taken to improve it.

Having described the basic contours of the writing workshop model, I can now turn back to the topic at hand: Lotus Chorus Workshop, an algorithmic “writing workshop simulator.” Of course, this phrase sounds rather cheeky; how could one simulate the intimacy and vulnerability of a writing workshop (not to mention the interplay of writerly egos)?

Indeed, Lotus Chorus Workshop is more of a send-up of the creative writing workshop than a verisimilitudinous reproduction. If there is a comedic effect in its sometimes-nonsensical suggestions or even in its text-based interface, which represents workshop participants with emoji (e.g., 🙄 and 🙋), this effect is not unintended. However, I also mean — unironically — for Lotus Chorus Workshop to simulate the way that a creative writing workshop can provide a nutrient-dense and nutrient-diverse substrate in which to grow a piece of literature.

## 2. Lotus Chorus Workshop

Lotus Chorus Workshop is a text-based interface written in Python. The human writer composes a poem sentence by sentence; after each sentence, Lotus Chorus Workshop provides the writer with feedback and asks for a revision. (On rare occasion it will be satisfied with a line and abstain from asking the writer to meddle with it.) Already this simulation of a creative writing workshop seems to have departed from what it allegedly simulates; a member of a creative writing workshop can expect their work, typically a full draft, to “be workshopped” only infrequently, certainly not in the moment-to-moment process of writing it. My original motivation — one that owes its existence to the isolation brought on by the first months of the Covid-19 pandemic — was to simulate, even in an exaggerated manner, the zesty sociality of the workshop experience. The more frequently that Lotus Chorus Workshop comments on the writer’s text, the less alone, perhaps, the writer feels.

Systems that provide feedback and suggestions to creative writers tend to provide just one type of feedback, whether by using a neural network to suggest the next sentence in a story (Roemmele & Gordon 2018), revising a slogan by fitting words into a syntactic template (Clark et al. 2018), or suggesting metaphorical connections (Gero & Chilton 2019). However, sophisticated the code of these writing-assistance tools may be, they are what we might call *unifunctional*. Spell check as well as common forms of predictive text software are also unifunctional. Spell check will draw a red line under a misspelled word; it will not from time to time critique one’s word choice. Predictive text will suggest the next word; it will not on occasion remark that an email has gone on long enough.

Other systems for providing feedback are multifunctional. When composing a poem using the iOS app Abra (Borsuk et al. 2015), the writer is constantly surprised when their text is transformed in a variety of ways (e.g., the poem may suddenly be filled with emoji, or words may disappear or change color). Less-artistic tools for writers may also be multifunctional. Grammarly (grammarly.com), for instance, attends to the text at a variety of levels. It may simultaneously point out that a certain sentence is ungrammatical, that a certain slang word is out of place in a business email, and that the text in general may be a bit too difficult to read.

Lotus Chorus Workshop is a multifunctional system. At present it possesses nineteen different functions, each providing a very different response to the human writer's input text. As I composed these functions, I have endeavored to make them diverse both in terms of the scale of the text upon which they comment (e.g., letters, words, phrases or clauses, or its entirety) and the content and tenor of their suggestions.

## 2.1. Critique Functions

**Simple Noun Comment** picks a noun at random from the input text and generates a response by picking an adjective at random from a list.<sup>1</sup> For instance, given the sentence:

Input: I awoke from the cave.

this function may reply:

Output: In my opinion, the word "cave" is a bit too salty.

or:

Output: What about the "broad cave"? Or "jazzy cave"?

**Wikipedia Rare Word Suggestion** picks a noun at random from the input text and generates a response by searching for this noun's entry on Wikipedia and suggesting a "rare word" from this page. A "rare word" here is defined as a noun or adjective that does not likewise appear in a large comparison corpus.

Input: The seer worries now about skin.

Output: This needs some more interesting language... when I see the word "skin" I think of "intercalary."

---

1. One such list includes all adjectives in the WordNet corpus (Fellbaum 1998).

**Ban Topics** chooses one or more predefined topics (e.g., “industrial” or “war”) and forbids it or them. Each topic possesses a list of search terms (e.g., for “industrial,” the list [“factory”, “machine”, “forge”, “electricity”...]). This function uses a vector-space language model (Pennington, Socher, and Manning 2014) to check words in the input sentence against these search terms. If a word in the input sentence is similar enough to several of these search terms according to the model (calculated as cosine similarity between the vectors, as is standard), this function will complain:

Input: A spider calculates the scalar of the house.

Output: “house”?...ugh...enough of this tedious architectural stuff!

**Ban Letter** forbids at random a letter from the set of letters that the input sentence uses more than a minimum number of times:

Input: One can find it grown in the igneous gift.

Output: Try getting rid of all the “i”’s. There are entirely too many.

**Critique Line Length** observes how many words are in the input sentence and suggest that this text be lengthened or shortened.

Input: One can find it grown in the igneous gift.

Output: This is 9 words long, but I think the perfect number of words would be...5.

**Syntactic Chunk Comment** parses the input sentence using the Berkeley Neural Parser (Kitaev, Cao, and Klein 2019), breaking it into chunks. This function filters out chunks that are too short or too long, chooses one of them, and comments on it with a random word or phrase:

Input: That old wallet photo could wreck a lot of wolves.

Output: “that old wallet photo”...I just feel like this needs some rhythm?

**Word Morph** picks a noun, adjective, or verb at random from the input sentence and suggests that the writer replace it with a word that begins with a specific letter and that contains a certain number of syllables. Sometimes the writer is also given a specific genre or author to emulate with this word choice:

Input: One can find it grown in the igneous gift.

Output: What if you swapped “gift” with a 5 syllable word that begins with “e” and that sounds like it could be from an old book about how a language works?

**Flip** suggests that the writer rewrite the input sentence so that it begins with a different word that is already in the sentence:

Input: A drill the dentist wears on a string.

Output: I'd move "dentist" to the beginning of the sentence.

**Ban Parts of Speech** randomly chooses one or more parts of speech used by the input sentence and forbids it or them:

Input: Running won't save water from falling.

Output: Try this again but without any gerunds or present participles.

**Demand Parts of Speech** notices (based on part-of-speech tagging) whether the input sentence employs, for instance, the future tense, the past tense, or a modal verb, suggesting one of them if it is not already present.

Input: Running won't save water from falling.

Output: Maybe you could use a word like "could" or "should" to entertain possibilities.

**Specify Noun** selects a noun from the input sentence and, also picking at random a geographical location from a list, commands the writer to make this noun more specific:

Input: A drill the dentist wears on a string.

Output: Too vague. Make this a specific drill from Ancient Rome.

**Suggest Rhyme** picks at random a noun or adjective from the input sentence and suggests that this word be made to rhyme with another noun or adjective already in the sentence:

Input: That old wallet photo could wreck a lot of wolves.

Output: Replace "wallet" with a similar word that rhymes with "wolves."

**Prefixification** picks at random a noun from the input sentence and suggests that it be replaced by a neologism formed by adding to this word a prefix (e.g., "auto," "meso", or "Euro-"). Occasionally this function will also suggest a suffix (e.g., "oid," "otrope," or "ette"):

Input: Running won't save water from falling.

Output: Instead of "water," what about "pseudo-waterite"?

**Google Inspiration** picks a noun at random from the input sentence, picks at random a word or phrase from a list, and suggests that the user searches both at once via Google (or, less commonly, another resource, such as the user's email).

Input: Running won't save water from falling.

Output: You need to get outside your own head. Why don't you try googling "water" and "downtown scene"?

**Figurificate** simply recommends that the sentence be revised to make use of one or two literary figures drawn from online databases (Burton, n.d.; "Literary Devices," n.d.). Its selection is random and does not depend upon the user's input:

This would be so much better if it had an eutrepismus and a charientismus.

**Meta-Critique** implies that the input text does not meet some definition of poetry or some specification of poetry's purposes, chosen at random from a list of such definitions and purposes from famous writers, including Wallace Stevens, Audre Lorde, and (in this case) Emily Dickinson:

Just so you know, poetry "should feel physically as if the top of one's head were taken off."

Subsequent invocations of this function become increasingly strident:

What is wrong with you? Don't you get that poetry should feel physically as if the top of one's head were taken off?

**Deconstructed Sestina** recommends that the writer end a sentence with a specific word chosen randomly from a list of words that are general-purpose and unassuming. This function selects some number of these words and rotates through them repeatedly. As with the Figurification and Meta-Critique functions, this function's output is not shaped by the input sentence:

Try ending with this word: "away."

And then, some lines later:

Try ending with this word: "fortune."

And then, some lines later:

Try ending with this word: "face."

And then, some lines later:

Try ending with this word: “away.”

And then, some lines later:

Try ending with this word: “fortune.”

Should the writer follow these suggestions, their poem will come to resemble the cascading repetition of a sestina.

**Chastise Uncommon Words** observes when the input text contains rare words (those that do not appear, or appear only once or twice, in a large corpus). It complains about them:

Input: I will animadvert against my infelicitous fate.  
Output: “Animadvert,” “infelicitous”...a little pretentious, no?

**Extend Line** implies that the input text is of sufficient quality but that the writer should keep going, extending it in some way. It does so based on the part of speech of the final word of the input text. Different parts of speech correspond to different words or phrases that this function uses to prompt the writer’s continuation. For instance:

Input: That night, dawn walked upon the field.  
Output: Yes! Keep going: “...walked upon the field, the kind...”

But:

Input: Take no water unless it be perfect.  
Output: Yes! Keep going: “...unless it be perfect as those...”

## 2.2. Consternation and Flowers

I have described the critique functions that Lotus Chorus Workshop has at its disposal. How are these functions fitted together?

As I have argued, a creative writing workshop is designed to provide feedback that is not just overabundant but possibly discordant, or at least not obviously consonant. To my knowledge, Lotus Chorus Workshop is the algorithmic assistant for creative writers that tries to mimic the cacophony of the workshop model. It is not just multi-functional but *polyvocal*, simulating the interplay of various voices.

Lotus Chorus Workshop achieves this effect via a meta-function that is in charge of making use of the various critique functions



described above. Each critique function is assigned to a participant in the Workshop, visualized as an emoji person, typically one who seems to express either eagerness to help (such as 🧑) or disappointment or disgust (such as 🙄). The human writer inputs a line:

That old wallet photo could wreck a lot of wolves.

And, typically, one or more of their emoji peers respond:

🧑: Too vague. Make this a specific photo from Byzantium.

🧑: Try this again but without any adjectives.

🧑: What about the “slipperd wallet”? Or “lugubrious wallet”?



🧑: I’d move “photo” to the beginning of the sentence.

While the human writer’s emoji peers do not explicitly contradict each other (i.e., they do not get into arguments, as sometimes happens in a workshop setting), they may do so implicitly, as when one praises a noun phrase and another complains about a word within it, or (as in the above example) one suggests an adjective and one advises eschewing adjectives. Even when these comments seem not to speak to each other at all, the user may feel obliged to integrate them in some way (e.g., “What sort of photo from Byzantium might have something to do with a ‘lugubrious wallet?’”).

Each time the user starts the program, the meta-function randomly selects some number (the number itself varies) of the nineteen critique functions to be used during that session; these are also randomly weighted so that some are more likely to be used than others. The meta-function also randomly determines the maximum number of critical responses to each of the user’s input sentences. Randomization also determines the likelihood that the Workshop will declare itself satisfied with a given input line, passing over the user’s words with (often tepid) praise instead of pausing to offer criticism. Individual functions too may behave differently each time the program is restarted. For instance, the Ban Topics function randomly chooses anew the topic or topics that it will ban, and the Demand Parts of Speech function randomly chooses several parts of speech to encourage from a list of them. These randomized parameters are meant to keep the user from feeling too certain of how Lotus Chorus Workshop will respond as well as to increase the “replay value” of the system, since different combinations of critique functions may work together to encourage poems with very different styles and tones. Lotus Chorus Workshop may ask the user to change the word “shell” so that it rhymes with “vase” and is also more “politically im-

portant”; or, based on the same input from the writer, it may command the writer to do this but also to remove all gerunds and move the word “shell” to the beginning the sentence. In general, writing with the Workshop becomes more challenging the greater the maximum number of comments it provides per sentence.

Lotus Chorus Workshop is a workshop simulator, but it is also a “chorus” in the sense that it is composed of a crowd that gives commentary (though perhaps this term implies more harmonious unity than the discord it often mutters). But what about “Lotus”? The creative writing workshop is a notoriously unnerving ritual; no doubt few writers, no matter how self-possessed, can silently endure a roomful of critics without at least a small spike in cortisol. In an attempt to maintain the user’s goodwill, Lotus Chorus Workshop will sometimes proffer a gift as recompense for a line’s revision. A lotus — a symbol of purity and calm — would make the perfect gift, perhaps even an antidote, but really any flower will do:

: For your efforts, please take this: 

### 3. Project Outcomes

Lotus Chorus Workshop is open-source software and is available in an online repository.<sup>2</sup> It is a modular platform; modifying it is as easy as writing another function that responds to the user’s input text. I invite others to make their own copy of Lotus Chorus Workshop and change how it works simply by their own critique functions (as well as by changing or deleting the ones I have written).

However, a workshop — whether a real one or a computational simulation — is not an end unto itself but a technique for producing and refining literature. Several of my poems written with the assistance of Lotus Chorus Workshop appeared in issue #28 of UK poetry venue *Blackbox Manifold* (Booten 2022). When presenting my work composed via Lotus Chorus Workshop, I do not exclude the computer-generated critical remarks. (One could imagine an empirical study that would test whether poems written with Lotus Chorus Workshop — *sans* its interlineated suggestions — are regularly different in some way from poems written without its help.)

Below I include a poem written with the assistance of Lotus Chorus Workshop, one of the three such poems that appeared in *Blackbox Manifold*. My inputs are prepended with “>”:



a γυμνάσιον

\*\*\*\*\*

INSTRUCTIONS:

- write a sentence of poetry
- receive feedback
- revise your sentence accordingly
- repeat
- type "quit" to quit

\*\*\*\*\*

>Could you go without checking your phone all winter?

👩: Try ending with this word: "nature."

>Could you go without checking your phone until winter, or does this violate your new nature?

👨: Well revised...now keep going, write a new sentence.

>Could you lie prone in a shallow crater without moving your head at all?

👩: Too vague. Make this a specific head from Illyria.

👩: This needs some cleverness...when I see the word "head" I think of "cephalization."

👩: What if you swapped "head" with a 2 syllable word that begins with "l" and that puts one in mind of the Apophthegmata?

>Could you lie still in a shallow crater with only thin bronze flaps to shield your cephalization?

👨: Good. Now write another sentence.

>In your dreams, when the helicopters list and fall, how fast are you?

👩: "Dreams"?...really?...enough of this damn romance stuff!


👨: Try getting rid of all the "h"'s. There are entirely too many.


>In your vale of REM, when the Kamovs list and sink, do you escape?

👨: See how your verse becomes more liberated?



>When you imagine movements in your mind, even asleep, your body experiences this as practice.

👩: Try ending with this word: "earth."


: Instead of “practice,” what about “practice-gate”?


: Replace “practice” with a similar word that rhymes with “movements.”

>To imagine movement is itself attunement of matter--- hence all those monks must close their eyes to spin the earth.

: You have earned this symbol of your openness: 

>Soldiers used to dress like toxic pheasants.


: What if you swapped “used” with a word that begins with “p”?

: Try this again but without any prepositions.


>Soldiers previously wore toxic pheasant padding.

: Good.


>Just now, while rubbing the newspaper, I was served an advertisement for tactical sweatpants.

: Not bad. Go on.

>Because of my background, I understood that this product did not exist, yet that by clicking it I could make it real.

: Too vague. Make this a specific product from Ancient Greece.

>Because of my background, I understood that this product---CozyGreaves, blue-gray and Kevlar-jersey blend-- did not exist, yet that by clicking it I could make it real.

: Well revised.

## 4. Designing for Cognitive Overload

A reader who is suspicious of the very notion of a “writing workshop simulator,” who still thinks that this concept could only be a joke, may have picked up on the number of times that my description of Lotus Chorus Workshop has noted that a critique function operates “randomly.” One’s peers in a writing workshop might offer suggestions that are poorly thought-out, insincere, or even malicious, but not purely aleatoric. Clearly what is missing from Lotus Chorus Workshop is a genuine capacity for aesthetic judgment. If it suggests that a line “could be a little more crisp,” this is not because it has decided that it is mushy. Its feedback is utterly capricious.

But how do we know that *clever* feedback — observations, critical statements, and suggestions generated by the minds of the well-meaning and the well-educated — is entirely better, or always better, than *random* feedback? Lotus Chorus Workshop cannot replace the critical estimations of an actual creative writing workshop. It cannot identify when a poem is maudlin, when it seems too close

in style to John Ashbery, or when a particular rhyme (“regret” with “baguette”) feels forced. It cannot offer a sincere diagnosis of a real problem. However, as I have suggested, the traditional workshop does not itself seem designed to produce the clarity of a diagnosis, since its participants can hold different opinions on every matter of importance. When a workshop adjourns, a writer may stagger into the twilight feeling only more confused, even a bit punch-drunk. While this may sound somewhat perverse, it is not, at least not in the case of poetry. The poet — or, to be less presumptuous, a certain kind of poet — must figure out a way to rise above mundane, boring language, which means that the poet’s cognitive apparatus must also find a way out of the rut of mundane, boring thought. With this in mind, one virtue of the traditional writing workshop at its most confusing is that it creates a situation of “distributed cognition” (Hollan et al. 2000); a group of people can think up more — and more diverse and difficult-to-reconcile — opinions about how a text should be revised than would have been possible for the writer to generate alone.

Lotus Chorus Workshop is designed to push the writer into a similar state of *cognitive overload*. It takes the solitary activity of writing and injects what Bakhtin (2008) would call “heteroglossia” and others would call “too many cooks in the kitchen.” It forces the writer to simultaneously attend to very different levels of the text, from the tone to the rhetorical structure to individual words, parts of speech, and even letters. Like a writing workshop, it makes the writing process more complicated by cluttering it with additional goals that may become even more challenging in light of each other. Unlike the human critic (or, for that matter, the human writer), this system does not feel the tug of reasonableness. It is free to be “random.”

As I have argued elsewhere (Booten 2021), computational systems that “make writing harder” by providing the writer with feedback can be seen as digital analogues of traditional verse forms. The form of the sonnet, for instance, makes writing a poem more difficult by forbidding the writer from using certain words that would not fit the rhyme scheme or the meter; the gambit of such a form is that embracing it — taking on the additional cognitive burdens enforced by its rules — goads the writer beyond the leaden gravities of common sense and linguistic custom. In what ways is writing with Lotus Chorus Workshop different from writing in the rhyming iambic lines of a sonnet? What is gained by the complexity of the digital interface, which is certainly less elegant and less portable than the sonnet’s simple “algorithm”?

A general difference: Lotus Chorus Workshop, taking its inspiration from the writing workshop, is designed to place the writer into a state of *cognitive overload*. It seeks to overwhelm the writer with *quantity* (the number of suggestions given), *diversity* (the very different types of suggestions), and *unpredictability* (the fact that

randomization occurs at several levels, determining which critique functions will be chosen for the duration of a poem's composition, which will "fire" after a given line, and exactly what sort of feedback a specific function will offer). Within mainstream Human-Computer Interaction research, the phrase "cognitive overload" has long carried a pejorative connotation, and designers often shape tools and systems to minimize the strain they place on cognitive resources such as working memory (see, for instance, Mackay 2000). For those who engage in weightlifting, however, "overload" is a positive word, since it is only by pushing one's body beyond what it may comfortably accomplish that it becomes stronger. Lotus Chorus Workshop takes this more "athletic" approach to writing. In fact, just as a weightlifter might gradually add weights to produce "progressive musculoskeletal overload," a poet might first use Lotus Chorus Workshop with only one or two of its critique functions in operation, over time adjusting the difficulty by increasing the number of functions from which the system will choose as well as the number of them that can respond to a given input line. (I have found myself adjusting these numbers depending on my mood and energy.)

A gym-goer should not frequent the bicep-curl machine unless they want their biceps to become larger. If a writer wants nothing more than a single, coherent piece of advice, they should probably seek out an individual mentor, not jump into the heteroglossic cacophony of a creative writing workshop. Likewise, a poet should avoid Lotus Chorus Workshop if their aesthetic goals do not match the sort of poetry that its feedback encourages. Though my own writing here has provided one illustration of what this sort of poetry might look like, it remains to be seen whether other poets using this system in their own practice might achieve very different aesthetic effects. Still, as I designed Lotus Chorus Workshop, I took inspiration — however loosely — from poets whose verse seems to both produce and be the products of cognitive overload;<sup>3</sup> it seems to me to be unlikely that Lotus Chorus Workshop would push the writer toward competing literary virtues, such as meditative equipoise<sup>4</sup> or an unadorned but confident plain style. Perhaps designing and building a different sort of computational writing-assistant, one that would encourage *deep* calmness or simplicity (neither of which is to say quietude) would not be an uninteresting goal. As it stands, however, the logic of increasingly-ubiquitous "human language technologies" does seem to promote a distinctly *stultified* simplicity. Google's autocomplete nudges our queries away from odd, idiosyncratic phrasing and "into more regular, economically exploitable linguistic subsets" (Kaplan 2014, 60). Likewise, Grammarly's suggestions (many of which the user can accept with a single click of the mouse) aim to snip and untwist the most gnarly sentences until they are straightforward and

---

3. For instance, English poet Keston Sutherland.

4. For instance, American poet Gary Snyder.

ergonomic. And LLMs such as OpenAI's GPT-3 and ChatGPT, for all their billions of parameters and the linguistic dexterity that these parameters afford them, seem most comfortable when generating bland, sensible opinions in a bland, sensible style. As these algorithms find new ways to nudge our own writing, we may have to work hard to be overwhelmed and to be overwhelmed in turn.

These LLMs now threaten to make writing as efficient and nearly-thoughtless as meal-kit services make cooking. Grammarly's forthcoming "GrammarlyGO" software, for instance, promises to "unblock your ideas and enable accelerated productivity for teams and individuals" by generating drafts based on "a prompt with a few words" (Grammarly, n.d.). Such promises should give us pause, since the importance of writing is not — or at least not merely — a matter of thought's fleet generation and communication. Writing also transforms thought, and it does so exactly because trying to put one's thoughts down on paper or screen allows the writer to discover *problems* — tensions between ideas, for instance, or between ideas and rhetorical goals; to solve these problems, the writer must allow her own thoughts and goals alike to be clarified, restructured, and metamorphosed (Scardamalia & Bereiter 1987; see also Emig 1977). A paradoxical sort of technology, writing can make thinking harder as well as slower. Any other technology that is *solely* devoted to making communication easier and faster cannot be in the service of "writing" in this sense.

As Bernard Stiegler (2020) has argued, we now must figure out how to use algorithmic technology "therapeutically," taking the same digital techniques that would make us blithely unthinking and using them to enable new forms of thought and reflection. In the design-based research presented here (an example of what Stiegler would perhaps consider "new organological craftsmanship" (218), the design of alternative instruments of thought), I have begun to explore one way of approaching this challenge: if writing allows us to think new thoughts by helping us to generate and solve problems that emerge within and between our sentences, one thing that algorithmic media can do for us is to multiply the number and variety of these problems, asking of us what we would be far too kind, too humane, to ask of ourselves. But, then again, this is not merely a question of algorithmic mechanism or interface design but, as Stiegler has also observed, of will and desire; any newfangled instrument profits nothing unless humans invest into it their own drives (239) — much the way that writers invest their desires to be writers into the rough ritual of the workshop. Yet does not this sort of psychic submission to "a master that is also a 'metre', a *metron*" (239) grow more difficult with the seductive release of each new consumer-grade tool that promises to eradicate from the act of writing any stress, any problem? As at the gym, laboring to stand beneath the overwhelm-

ing weight, we may have to learn to identify the correct form of suffering, seek it out, and take satisfaction from it.

## References

- Bakhtin, Mikhail.** 2008. *The Dialogic Imagination*. Edited by Michael Holquist, translated by Caryl Emerson and Michael Holquist. Austin, TX: University of Texas Press.
- Booten, Kyle.** 2022. "Lotus Chorus Workshop." *Blackbox Manifold* 28 (Summer).
- Booten, Kyle.** 2021. "Making Writing Harder: Computer-Mediated Authorship and the Problem of Care." *electronic book review* (September). <https://doi.org/10.7273/qr36-cw86>.
- Borsuk, Amaranth, Kate Durbin, and Ian Hatcher.** 2015. *Abra: a living text*, iOS ed., V. 1.0.
- Burton, Gideon O.** n.d. "Silva Rhetoricae." Accessed October 20, 2022. <http://rhetoric.byu.edu>.
- Clark, Elizabeth, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith.** 2018. "Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories." *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, 329-340. Tokyo. Association for Computing Machinery.
- Emig, Janet.** 1977. "Writing as a Mode of Learning." *College Composition and Communication*. 28, no. 2: 122-128.
- Fellbaum, Christiane.** 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.
- Gero, Katy Ilonka and Lydia B. Chilton.** 2019. "Metaphoria: An Algorithmic Companion for Metaphor Creation," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1-12. Glasgow.
- Grammarly.** n.d. <https://www.grammarly.com/grammarlygo>. Accessed April 20, 2023.
- Hollan, James, Edwin Hutchins, and David Kirsh.** 2000. "Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research." *ACM Transactions on Computer-Human Interaction* 7, no. 2 (June): 174-196.
- Kaplan, Frederic.** 2014. "Linguistic Capitalism and Algorithmic Mediation." 2014. *Representations* 127, no. 1: 57-63.
- Kitaev, Nikita, Steven Cao, and Dan Klein.** 2019. "Multilingual Constituency Parsing with Self-Attention and Pre-Training." *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3499-3505. Florence.
- "Literary Devices."** n.d. Accessed October 20, 2022. <https://literarydevices.net/>.
- Mackay, Wendy.** 2000. "Responding to Cognitive Overload: Co-adaptation Between Users and Technology." *Intellectica: Revue de l'Association pour la Recherche Cognitive*, no. 30: 177-193.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning.** 2014. "GloVe: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532-1543. Doha. Association for Computational Linguistics.
- Roemmele, Melissa and Andrew S. Gordon.** 2018. "Automated Assistance for Creative Writing with an RNN Language Model." *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*, 1-2. Tokyo.
- Scardamalia, Marlene and Carl Bereiter.** 1987. "Knowledge Telling and Knowledge Transforming in Written Composition." In *Advances in Psycholinguistics, Volume 2: Reading, Writing, and Language Learning*, edited by Sheldon Rosenberg, 142-75. Cambridge, UK: Cambridge University Press.
- Stiegler, Bernard.** 2020. *Nanjing Lectures 2016-2019*. Translated by Daniel Ross. London: Open Humanities Press.