BERTian Poetics:
Constrained Composition with Masked LMs

Christopher Akiki and Martin Potthast
Text Mining and Retrieval Group
Leipzig University
Germany
{christopher.akiki,martin.potthast}@uni-leipzig.de

Abstract

Masked language models have recently been interpreted as energy-based sequence models that can be generated from using a Metropolis–Hastings sampler. This short paper demonstrates how this can be instrumentalized for constrained composition and explores the poetics implied by such a usage. Our focus on constraints makes it especially apt to understand the generated text through the poetics of the OuLiPo movement.

1   Introduction

The cultural capital (Bourdieu, 2002) of poetry has rendered it fertile ground for computational approaches, making poetry generation "an established research field", recently surveyed by Gonçalo Oliveira (2017). This field spans a spectrum between two different perspectives which we find fitting to call, *machine-as-poet* and *user-as-poet*. The present work is aligned with the latter.

Machine-as-poet approaches encode *poeticity* (Jakobson, 1981) in a corpus and train a system to reproduce the prosodic and stylistic structure encoded within it (Ghazvininejad et al., 2017; Lau et al., 2018). User-as-poet approaches aim to produce tools that a *poet-programmer* (Morris, 2012) or *operator* (James, 2006) might use in their creative endeavors (Uthus et al., 2019; Zhipeng et al., 2019). Both approaches capitalize on recent advances in neural text generation, yet they exclusively involve auto-regressive language models.

The contribution of this paper is twofold: First, in Section 2, we implement[1] the Metropolis–Hastings sampler introduced by Goyal et al. (2021) and use it to explore constrained composition using a masked language model.[2] Second, in Section 3, we reflect on useful vantage points through which to critically understand our approach, specifically how it might be enacting oulipian constraints (Symes, 1999; James, 2006).

2   Constrained Composition

Transformers (Vaswani et al., 2017) have allowed the scaling up of unconditional language models to unprecedent magnitudes (Brown et al., 2020). Such causal models can be used to generate text by defining a probability $P(\mathbf{X})$ over sequences that can be sampled from auto-regressively. In practice, however, they prove unwieldy (Holtzman et al., 2020) and awkward to steer, rendering controllable generation a challenging open question (Weng, 2021), and out of reach for a flexible user-as-poet approach.

---

[1] https://github.com/webis-de/ML4CD-21

[2] Our implementation uses the base version of English roBERTa (Liu et al., 2019) off the Hugging Face shelf (Wolf et al., 2020), but the ideas introduced here are agnostic to masked language model and language.

Beyond those lines, the **unforeseen.**
The worst will never be predicted.
Beyond those lines, the **unforeseen.**
The unexpected will never be possible.

I cannot leave you. You have just passed away,
I cannot believe it! I have finally found freedom.

Figure 1: Explicit prompts in bold. **Left**: Lipogram through implicit prompting that filters out all vocabulary tokens containing the letter "a". **Right**: Prompting the masked language model with a left context.

Instead, we turn our attention to the findings of Goyal et al. (2021), who develop a tractable sampling scheme for masked language models. Even though they do not explicitly model sequences, masked language models are interpretable as energy-based sequence models that can be sampled from using the stationary distribution $P(\mathbf{X})$ of a Metropolis–Hastings Markov Chain.

In reality, the distribution being sampled from is $P(\mathbf{X} \mid l)$: a probability over all sequences of length l. This constitutes the first constraint that the operator needs to set in advance and the only obligatory one. A prompting context c can also be specified so that we sample from $P(\mathbf{X} \mid l, c)$. However, unlike auto-regressive models, this prompting context need not be causal but can span any subset of tokens. We refer to the general case of a non-contiguous context as **perforated prompting**. More generally, we introduce the following distribution over sequences and a list of poetic constraints $C_{1:n}$:

$$P(\mathbf{X}, C_{1:n}) = \prod_{i=1}^{n} P(C_i) \cdot P(\mathbf{X} \mid C_{1:n}), \tag{1}$$

which can be read as a product of experts (Hinton, 1999), each constraining an aspect of the sequence. These constraints—be they syntactic,[3] semantic, or prosodic[4]—can be enacted in logit space on an arbitrary subset of the sequence's tokens by setting the appropriate logits to $-\infty$ through any of the following operations at every sampling step (see Figure 1 for examples):

- **Explicit prompting**: Restricts the vocabulary of the model to a chosen token.
- **Implicit prompting**: Restricts the vocabulary of the model to satisfy a constraint.

3 Poetics

Having defined all the constraints, as well as a seeding sequence,[5] the operator lets the sampling process run—potentially infinitely—which then proceeds to enumerate all token combinations of the model's vocabulary. This sampling process is modulated by knowledge gleaned from an immense training corpus, making certain combinations more likely than others.

The randomness inherent to sampling aligns this work with **aleatory poetics**, as the operator "deliberately engages with chance as a compositional principle" (James, 2012), specifically as prompts make the starting text underdetermined by design, with the final text having a "determinate final form" (ibid. 2012). Another useful angle is what theorists refer to as **computational poetry**, whose defining feature is a procedure or algorithm that "precedes and determines the poem" Morris (2012) and sees the poet-programmer step back and attend to information flows (ibid. 2012). Given the central role of the machine in our poetic endeavor, the latter is also aligned with **conceptual poetry**, as the creation modalities are at least as important as the ideas they attempt to express (Perloff, 2012a). The universe that the generated text is allowed to inhabit and explore has to be predetermined by the operator, so that "all of the planning and decisions are made beforehand and the execution is a perfunctory affair. The idea becomes a machine that makes the text" (Goldsmith, 2007).

Such programmatic principles notably modulate the work of the French **OuLiPo** movement, who use constraints to understand, "explore and expand the field of literature" (Poucel, 2012). In stark opposition to—and rejection of—the surrealist practice of automatic writing[6] and the debilitating openness of contemporary writing, oulipian writing sees constraints as "a generative tool that enables a conflict necessary for the renewal of poetic form" (Deming, 2009). Such conflicts fuel a "productive friction between the constraining algorithm and the author's desire for meaning" (James, 2006), allowing an artist to "maximize their options through minimizing their choice" (Symes, 1999).

---

[3]e.g.: *Lipogram*, a constraint which forbids the use of a given letter.

[4]e.g.: *Bouts-Rimés*, a literary game where a sequence of rhymes has to be expanded into a poem.

[5]The seeding sequence can also consist of `<mask>` tokens.

[6]This begs the facetious question, is GPT-3 (Brown et al., 2020) a surrealist?

4 Ethics

Concerns about the mechanistic aspects of the written word can be traced back to Plato's *Phaedrus* (Plato, 1972). The field of neural text generation can only exponentiate these anxieties. Without due consideration and reflected usage, the large language models we have come to wield so readily can only calcify our existing biases, and potentially introduce others. The stochastic parrots introduced by Bender et al. (2021) stand to cause damage, through an intertextual pastiche of all that is ugly on the internet. This intertextual view of the generated text, alluded to but not named in Bender et al. (2021), adds one final critical layer to our poetic enterprise, that of **found poetry** (Perloff, 2012b), except that the recontextualized collage of others' words is mediated within an impenetrable latent space through the weights of a neural network.

References

Bender, E. M., T. Gebru, A. McMillan-Major, and S. Shmitchell (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, New York, NY, USA, pp. 610?623. Association for Computing Machinery.

Bourdieu, P. (2002). *The Forms of Capital*, Chapter 15, pp. 280–291. John Wiley & Sons, Ltd.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), *Advances in Neural Information Processing Systems*, Volume 33, pp. 1877–1901. Curran Associates, Inc.

Deming, R. (2009, 12). Constraints as Opposed to What?: A Philosophical Approach to the Values of Constrained Writing. *Poetics Today 30*(4), 653–668.

Ghazvininejad, M., X. Shi, J. Priyadarshi, and K. Knight (2017, July). Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada, pp. 43–48. Association for Computational Linguistics.

Goldsmith, K. (2007). Paragraphs on conceptual writing. *Poetry Foundation*.

Gonçalo Oliveira, H. (2017, September). A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, Santiago de Compostela, Spain, pp. 11–20. Association for Computational Linguistics.

Goyal, K., C. Dyer, and T. Berg-Kirkpatrick (2021). Exposing the implicit energy networks behind masked language models via metropolis–hastings.

Hinton, G. E. (1999). Products of experts. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, Volume 1, pp. 1–6. IET.

Holtzman, A., J. Buys, L. Du, M. Forbes, and Y. Choi (2020). The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jakobson, R. (1981). *What is Poetry?*, pp. 740–750. De Gruyter Mouton.

James, A. (2006). Automatism, arbitrariness, and the oulipian author. *French Forum 31*(2), 111–125.

James, A. (2012). Aleatory poetics. In R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, P. Rouzer, H. Feinsod, D. Marno, and A. Slessarev (Eds.), *The Princeton Encyclopedia of Poetry and Poetics: Fourth Edition* (4 ed.)., pp. 31–34. Princeton University Press.

Lau, J. H., T. Cohn, T. Baldwin, J. Brooke, and A. Hammond (2018, July). Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 1948–1958. Association for Computational Linguistics.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). Roberta: A robustly optimized bert pretraining approach.

Morris, A. (2012). Computational poetry. In R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, P. Rouzer, H. Feinsod, D. Marno, and A. Slessarev (Eds.), *The Princeton Encyclopedia of Poetry and Poetics: Fourth Edition* (4 ed.)., pp. 288. Princeton University Press.

Perloff, M. (2012a). Conceptual poetry. In R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, P. Rouzer, H. Feinsod, D. Marno, and A. Slessarev (Eds.), *The Princeton Encyclopedia of Poetry and Poetics: Fourth Edition* (4 ed.)., pp. 292. Princeton University Press.

Perloff, M. (2012b). Found poetry. In R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, P. Rouzer, H. Feinsod, D. Marno, and A. Slessarev (Eds.), *The Princeton Encyclopedia of Poetry and Poetics: Fourth Edition* (4 ed.)., pp. 503–504. Princeton University Press.

Plato (1972). *Plato: Phaedrus*. Cambridge University Press.

Poucel, J.-J. (2012). Oulipo. In R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, P. Rouzer, H. Feinsod, D. Marno, and A. Slessarev (Eds.), *The Princeton Encyclopedia of Poetry and Poetics: Fourth Edition* (4 ed.)., pp. 987–988. Princeton University Press.

Symes, C. (1999). Writing by numbers: Oulipo and the creativity of constraints. *Mosaic: An Interdisciplinary Critical Journal 32*(3), 87–107.

Uthus, D., M. Voitovich, R. Mical, and R. Kurzweil (2019). First steps towards collaborative poetry generation. In *NeurIPS Workshop on Machine Learning for Creativity and Design 3.0*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Red Hook, NY, USA, pp. 6000?6010. Curran Associates Inc.

Weng, L. (2021). Controllable neural text generation. *lilianweng.github.io/lil-log*.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, pp. 38–45. Association for Computational Linguistics.

Zhipeng, G., X. Yi, M. Sun, W. Li, C. Yang, J. Liang, H. Chen, Y. Zhang, and R. Li (2019, July). Jiuge: A human-machine collaborative Chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy, pp. 25–30. Association for Computational Linguistics.