A Leap of Creativity: From Systems that Generalize to Systems that Filter

Porter Glines, Brandon Biggs, and **Paul M. Bodily**
Department of Computer Science
Idaho State University
921 S. 8th Ave, Pocatello, ID 83209 USA
glinport@isu.edu, biggbran@isu.edu, bodipaul@isu.edu

Abstract

In his work "Mere Generation: Essential Barometer or Dated Concept?", Ventura (Ventura 2016) categorizes creative processes along a spectrum of increasing creativity. While the spectrum provides insight into the dimensions through which creativity can be augmented, it does not of itself provide insights into how to advance a system through these dimensions. In this paper, we present some theoretical and practical insights on advancing along one commonly problematic rung of this ladder, namely from a system that exhibits *generalization* (i.e., the ability to generalize beyond an inspiring set) to a system that exhibits *filtration* (i.e., the ability to self-evaluate and filter results). One potential challenge in this transition is that filtration requires having a sufficiently large number of solutions to filter from the generalizing model. We propose that one solution to this problem is achieved not through increasing the size of the inspiring set (an obvious solution that brings additional problems), but rather through amplifying the generalization of the system to produce a greater set of novel artefacts to filter. We compare a new version of a system, NhMMonic, for generating creative mnemonic devices with a new conceptualization model that allows greater generalization. We demonstrate how filtration, which was not possible in the early version of NhMMonic, only becomes feasible with the more generalizable model.

Introduction

The field of Computational Creativity (CC) has been supported in its quest by several significant contributions in the domain of CC theory. One such contribution exists in Ventura's spectrum of creative systems (Ventura 2016). This spectrum suggests that there exist at least seven different levels along the path towards computational creativity including levels such as randomness, memorization, generalization, and filtration (see Figure 1). Ventura asserts that along this spectrum, real computational creativity starts at least as early as generalization with filtration representing perhaps a conservative threshold.

While this spectrum is useful for measuring the progress of applied CC systems, it leaves two important questions unanswered:

1. *For each level of the spectrum, what challenges are CC systems likely to encounter?*
2. *What suggestions can be made to overcome those challenges?*

Answers to these questions would provide a way to actualize the spectrum into a guide for augmenting the creativity of computational systems.

Our motivation in considering these issues came about in the context of our previous work using constrained Markov models to generate mnemonic devices (Bodily, Glines, and Biggs 2019). Markov models are an example of a generalizing model. The application of constraints to Markov models represents the act of filtration. In applying constraints to generate mnemonic devices, it frequently occurred that no satisfying solutions could be found.

The purpose of this paper is to provide answers to two questions stated above with specific regard to systems that have achieved the level of *generalization* and are attempting to make the "leap" to the level of *filtration*. This step is of interest as it marks the transition from a budding creative system to an intentionally creative system. This leap is significant in light of the fact that of the last four levels of the spectrum—where true creativity is said to emerge—this is the first step.

Generalization systems produce artefacts using an internal *conceptualization*—a model which embodies an understanding of a domain and allows for the creation of artefacts that belong to the domain (Ventura 2017). Examples of conceptualizations include using long short-term memory models for music generation (Nayebi and Vitelli 2015), neural networks for visual art (Norton, Heath, and Ventura 2013), and Markov processes for music and text generation (Pachet, Roy, and Barbieri 2011; Barbieri et al. 2012).

One particular challenge we have repeatedly observed in the development of CC systems at this level is the challenge of dealing with diminishing solution spaces. This problem arises commonly when attempts are first made to add filtration to a generalization system because filtration by definition implies the reduction of a system's solution space. The purpose of the filtration step is to equip the system with self-evaluative capabilities for restricting the artefacts it generates based on measurements of fitness. However, a well-known trade-off arises: stricter filtering leads to better, but

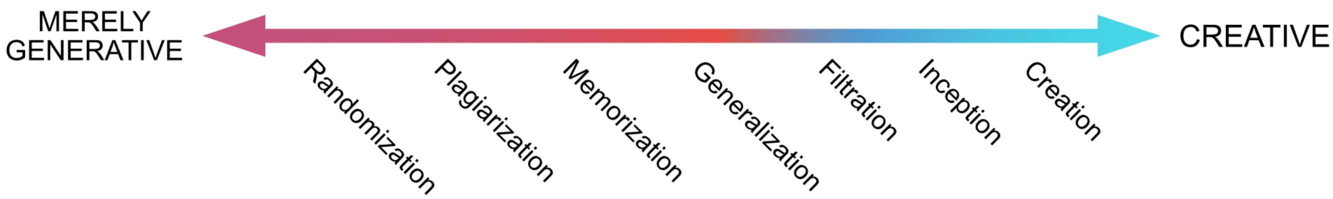

Figure 1: Ventura's (2016) spectrum of creative systems provides a means by which to measure the progress of a system towards becoming creative. Characterizing challenges and solutions that are specific to each level in the spectrum helps to actualize the spectrum into becoming a guide for building more creative systems.

fewer results. In some cases the results are so few that it becomes difficult to justify that the system is capable of generating anything, let alone artefacts that are novel. How can systems overcome this challenge?

A simple solution for increasing the solution space is to simply increase the size of the inspiring set. For many conceptualizations of CC systems this alone will increase the overall throughput of the system, and often increases the generalizability of the system as well. However, for most domains, finding a larger inspiring set ranges from being impractical to an impossibility. What more practical solutions exist?

We propose and illustrate through example how increasing the generalizability of a generalization system through abstraction and regularization can increase the solution space without requiring a larger inspiring set. Well-known methods exist for generalization of most conceptualization models used for CC systems, including L1 and L2 regularization for neural networks, shortening the Markov window length in Markov processes, generalizing the fitness function for genetic algorithms, and abstracting rules in rule-based systems. Through regularization and abstraction, a system is able to better leverage the knowledge in an inspiring set in order to increase the solution space.

In demonstrating the impacts of abstraction and generalization, we comparatively consider the performance of two models: a less abstract model (CoMP) and a more abstract model (CHiMP). We assess the ability of each model to intentionally produce *novel* artefacts. We choose to focus explicitly on the creative attribute of *novelty*—setting aside the attributes of value and intentionality—inasmuch as it is the attribute of creativity most directly relevant to our discussion (Ritchie 2007; **?**). We discuss the impacts of generalization on value in the discussion section below.

Methods

NhMMonic (Bodily, Glines, and Biggs 2019), is a CC system designed to generate mnemonic devices. At its heart, NhMMonic uses a constrained Markov process (CoMP) for its conceptualization model. This constrained Markov process allows for the combination of a (non-hidden) Markov process (e.g., trained on words) and a set of unary constraints (e.g., word-starts-with constraints) such that the model is able to generate constraint-satisfying sequences according to Markovian probabilities (Pachet, Roy, and Barbieri 2011). In previous work we demonstrated through qualitative surveys the strength of this model (particularly at higher Markov orders) for generating effective mnemonic devices. A byproduct of our analysis revealed that for many mnemonic device problems, the addition of constraints (i.e., filtering) resulted in NhMMonic being incapable of finding satisfying solutions despite being trained from relatively large inspiring sets.

A known method for increasing the generalization of Markov models is through the introducing of an abstract hidden layer resulting in a model known as a *hidden* Markov process. Direct dependencies between adjacent observed sequence elements are dissolved in the hidden Markov process, allowing for greater decoupling between sequence elements. This generally results in hidden Markov processes having significantly higher expressivity with respect to their *non-hidden* counterparts.

To combat the challenges facing NhMMonic with respect to a diminishing solution space, we designed a new conceptualization model for the system that combines hidden Markov processes with constraints in much the same way that constrained Markov processes combined *non-hidden* Markov processes with constraints (Glines, Biggs, and Bodily in press). The resulting model is called a constrained Hidden Markov process (CHiMP) which is visualized in Figure 2. The CHiMP model was chosen under the hypothesis that increased abstraction, resulting in increased generalization, would lead to a significantly larger solution space.

In implementing a filtration system, it is apparent that a large solution space is needed. Using two hypothetical models A and B (seen in Figure 3) we illustrate the restriction that solution space imposes on a system's ability to step from a generalization system to a filtration system. Model A fails to have a solution space after filtering and thus remains a conceptualization for a generalization system. Model B, however, has a larger beginning solution space β due to an increase in the model's ability to generalize the inspiring set.

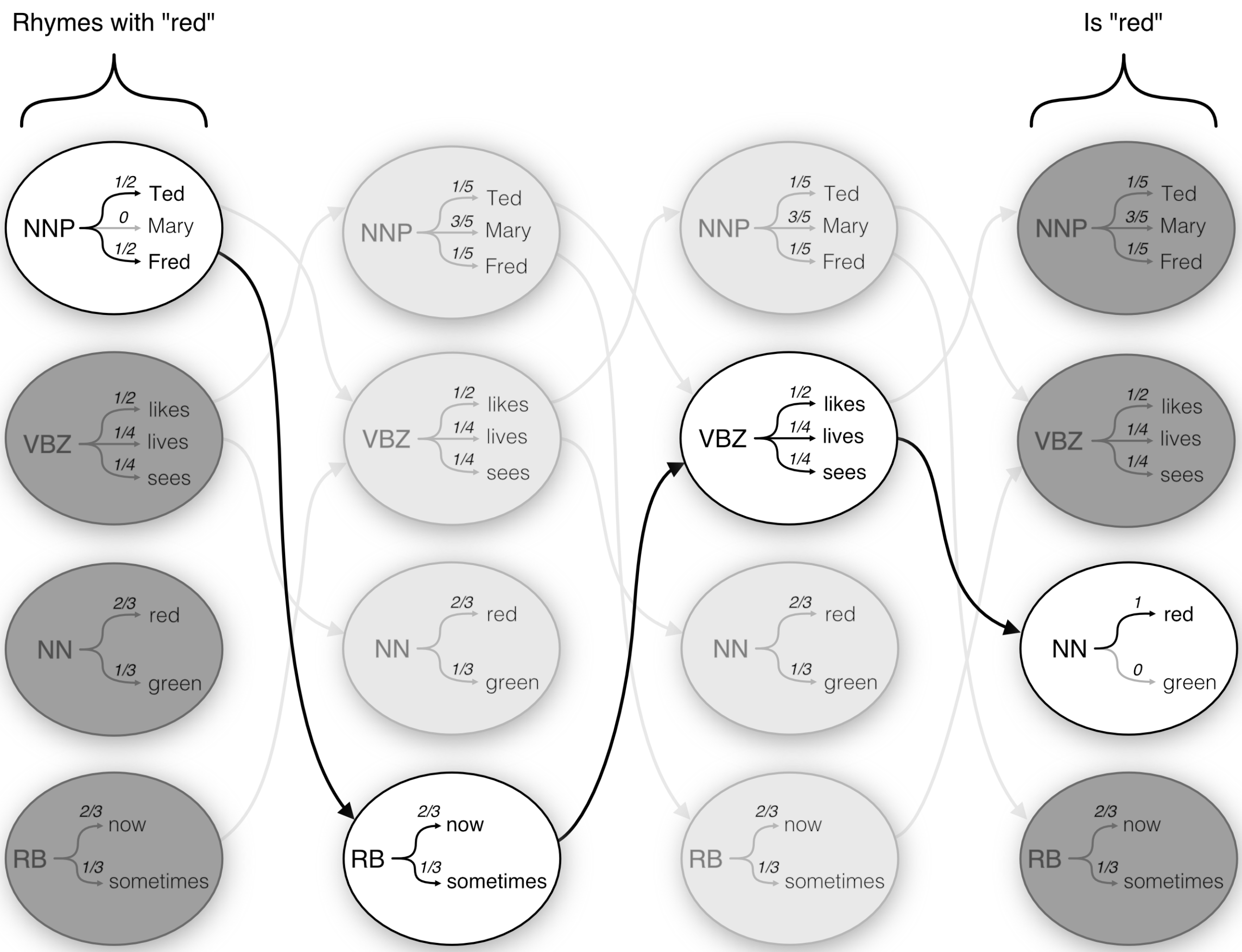

Figure 2: A high-level schematic of a constrained hidden Markov process (CHiMP) of length 4 constrained so that the last word is "red" and the first word rhymes with "red". Each column represents a position in the sequence to be generated. Each node represents a hidden state (i.e., part-of-speech) and a probability distribution for the observed states (i.e., words) that can be generated from that hidden state. By pruning observed states that are disallowed by constraints and then adjusting probabilities to maintain arc-consistency, the resulting model generates constraint-satisfying solutions with probability relative to the original probability distribution (Glines, Biggs, and Bodily in press). Hidden states pruned directly from applying constraints are indicated by dark grey nodes and states pruned during arc-consistency are indicated by light grey nodes.

Thus model B has a usable solution space β' after filtering and can be categorized as a filtration system.

Results

In demonstrating the increased generalization (and hence increased solution space) of CHiMP over CoMP, we compared the results of each model trained on the Corpus of Contemporary American English (COCA) (Davies 2009) and provided the same set of constraints. In particular, we selected training sets from the 2012 fiction portion of COCA and constrained each model to only output sequences in which the first letter of each word began with the same letter (e.g., a tongue-twister). We chose this problem because it represents a fairly general example of constrained sequence generation that is easily adapted to sequences of varying lengths. Results are averaged over 26 instances of the problem with each instance having constraints defined with a different letter of the English alphabet.

Some qualitative results are shown in Figure 4. It should be noted that within the subset of 40 sequences generated by CHiMP, no duplicate or similar solutions where present; whereas 6 sequences were duplicates (or very similar) in the subset generated by CoMP.

We examined the effect of changing the sentence/model length on the novelty of the system in terms of the total number of unique solutions capable of being generated by each model (see Figure 5). As the sentence length increases, so too do the number of constraints on the sequence to be generated. In the abstracted CHiMP model, this is inconsequential; the model can afford to make restrictions at the observed node that do not affect transitions between sequence positions (which are isolated in the hidden layer). Only occasionally do a sufficient number of pruned states combine to require the pruning of a hidden state node, but such is a relatively rare occurrence.

By contrast, the effects of increased sentence length on the CoMP model are severely limiting. Each added position would typically add a number of novel unique solutions *if it did not come with the addition of a new constraint*. The newly constrained position has direct influence on previous observed sequence states and thus pruning values from the domain of these variables directly results in the removal of transitions between adjacent sequence positions. This results in a relatively slow growth in the solution space as sentence length grows.

The increase in the CHiMP model appears to be exponential owing to the multiplicative effect achieved by maintaining large domains for adjacent variables in the hidden layer.

Similar trends in the impact on novelty are manifest when we vary the training set size, keeping sentence length constant (see Figure 6). We see that the size of the solution space for the CHiMP model increases exponentially. The CoMP model also appears to have some slightly exponential growth, but at a significantly lower rate. This is again what we would expect to see. Increasing the training set size (when such is a possibility) still has a more significant impact on CHiMP than on CoMP model.

The results shown in Figures 5 and 6 suggest that CHiMP, with respect to CoMP, facilitates exponentially more novelty. The solution space of the CoMP model is by definition a subset of the solution space of the CHiMP model, and for most training and constraint sets will be a substantially smaller subset. It is expected that of the novel results produced by CHiMP, some will have higher value than the solutions shared by both models. Because the CHiMP model abstracts to a more significant degree from the training set than the CoMP model, we might expect a greater portion of the novel solutions to be of lower value. The suggestion from qualitative results shown in Figure 4 is that there is no obvious degradation of value. However, we do not currently have results to fully assess the extent to which value degrades (or doesn't). In any case the expressivity of the CHiMP model enables a simple solution: introduce new or stricter filtering by increasing the number and stringency of constraints.

Discussion and Conclusion

In progressing from a generalizing system to a filtration system, our results provide meaningful insight into two important questions relating to Ventura's spectrum of creative systems:

1. *For the filtration level of the spectrum, what challenges are CC systems likely to encounter?*
2. *What suggestions can be made to overcome these challenges?*

A significant challenge for CC systems attempting to transition to a filtration system is as more constraints (or filters) are put on the system, the solution space diminishes to the point of being too small to filter. As demonstrated in the CoMP model (Figure 5), the insufficient solution space prevents being able to apply more constraints and filters to produce higher quality artefacts.

The problem is not specific to our results or to Markov models. Filtering, by nature, reduces the solution space. As shown in Figure 3, any CC system with low generalization may fail to have a usable solution space after filtering.

Greater generalization can address the aforementioned problem. We see from our results that our model with greater generalization, CHiMP, excels in solution space size even as constraints are added (see Figure 5). The primary difference between CoMP and CHiMP is an added layer of abstraction in CHiMP that affords greater generalization. The solution to a diminished solution space is to increase the level of abstraction in the model. This increases the generalization ability of the model and results in a solution space substantial enough to "survive" filtering.

Increased constraints allow for greater creativity and quality because the system can use constraints to explicitly articulate and enforce the system's goals and intentions. For example, in Markov models, increasing the Markov order (a form of adding more constraints) significantly improves the coherency of natural language, but the solution space is heavily diminished. With the CHiMP model, the solution space is sufficiently enlarged to avoid these devastating consequences to the solution space. Besides changes to the Markov order, other possibilities open up for using constraints to filter results to further improve quality, including

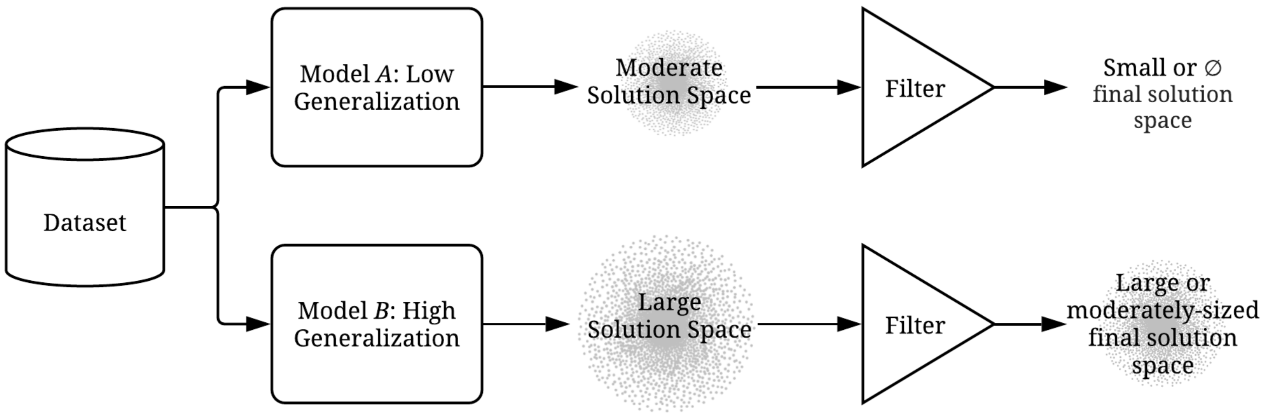

Figure 3: The application of filters on two hypothetical models (A and B) demonstrates the requirement for larger solution spaces (increased generalization) in order to endure filtering with a usable solution space. Model B has a usable solution space after filtering; thus the model has moved further along in the spectrum from generalization to filtration.

CoMP Tongue Twisters:

late last light levels like lady
Diaz did dinosaurs died dell drove
max mowed my mother made my
language lessons last look little lamb

CHiMP Tongue Twisters:

queen Quanhe quite quiet queasy qualified
flower facing forward for from forester
free feeling facing followed free fate
every educated Elizabeth expected Erika enchanting

Figure 4: Example results from generating 6-length tongue twisters (i.e., alliterative constraints) from both the CoMP and CHiMP models. Both models were trained on 10K sentences. Results are chosen from a randomly selected subset of 40 sequences from each model. The quality of tongue twisters is roughly equivalent between both models (both poor), but the CHiMP model is capable of generating exponentially more solutions. This suggests that increasing the Markov order in the CHiMP model (as an example of more stringent constraints) will have far less deleterious affects on the solution space as compared to a similar increase in the CoMP model.

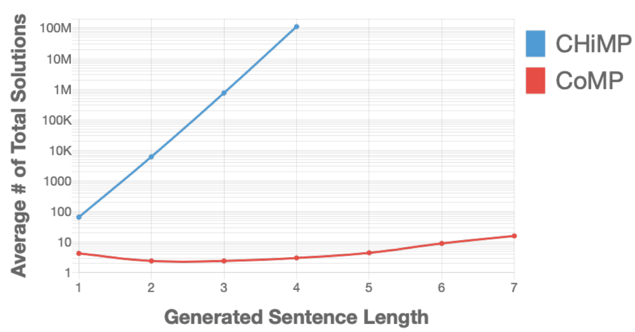

Figure 5: The effects of sequence length on the number of total solutions generated by each model with a fixed training set size of 300 sentences. Both models are constrained such that each word in a sequence starts with the same letter; counts of total solutions are averaged over 26 runs (each run using a different letter from the English alphabet). We see that as the sequence length increases, total solutions for the CHiMP model increases exponentially (given the logarithmic scale) whereas the CoMP model stagnates.

Proceedings of the 11th International Conference on Computational Creativity (ICCC'20)
ISBN: 978-989-54160-2-8

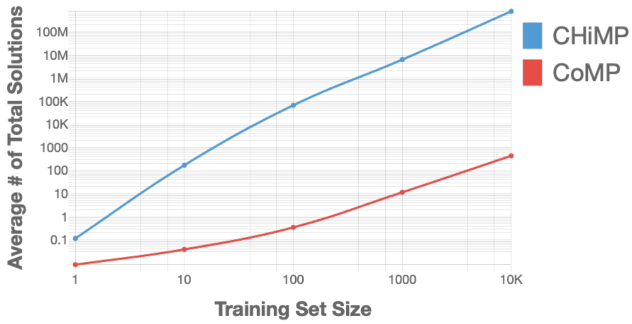

Figure 6: The effects of training corpus size (number of training sentences) on the number of total solutions generated by each model with a fixed sequence length of 3. Both models are constrained such that each word in a sequence starts with the same letter; counts of total solutions are averaged over 26 runs (each run using a different letter from the English alphabet). The total solutions of both models increase in an almost parallel way; however, at 10K training sentences, CHiMP well exceeds 100M total solutions which contrasts CoMP at 1000 total solutions.

semantic constraints, structural constraints, and even more complex $n-ary$ constraints. It is also often the case that constraints can be easily described in human-interpretable language, enabling the system to provide framing for its creative behavior, contributing to an increased perception of creativity in CC systems (Colton 2008).

It is important to acknowledge the negative consequences of increasing the generalization in a learning model. In particular, generalization decouples dependencies between variables which can result in a loss of information during variable assignment. For example, generalizing to a hidden Markov model takes a significant toll on language coherence. In short, the novelty achieved by generalization comes with a trade-off in value. We hypothesize that this deterioration can be offset in the application of filters to preserve the information lost. We plan to examine this issue in future work.

Through developing a system (CHiMP) that more effectively achieves filtration, we have discovered insights into the challenges present in the leap from *generalization* to *filtration* and how to overcome them. The challenge of diminishing solution spaces can be overcome by amplifying the generalizing ability of the system through abstraction. Having realized the leap from generalization to filtration, the community is now poised to address the challenge of making the subsequent leaps along Ventura's spectrum of creative systems, advancing past filtration into *inception* and ultimately *creation*.

References

Barbieri, G.; Pachet, F.; Roy, P.; and Esposti, M. D. 2012. Markov constraints for generating lyrics with style. In *Proceedings of the Twentieth European Conference on Artificial Intelligence*, 115–120.

Bodily, P. M.; Glines, P.; and Biggs, B. 2019. "She Offered No Argument": Constrained Probabilistic Modeling for Mnemonic Device Generation. In Kazjon Grace, Michael Cook, D. V. M. L. M., ed., *Proceedings of the 10th International Conference on Computational Creativity*, 81–88. Charlotte, North Carolina: Association for Computational Creativity.

Colton, S. 2008. Creativity Versus the Perception of Creativity in Computational Systems. *Proceedings of the AAAI Spring Symposium on Creative Systems*.

Davies, M. 2009. The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics* 14(2):159–190.

Glines, P.; Biggs, B.; and Bodily, P. in press. Probabilistic generation of sequences under constraints. In *Proceedings of the First Intermountain Engineering, Technology, and Computing Conference*.

Nayebi, A., and Vitelli, M. 2015. GRUV: Algorithmic Music Generation using Recurrent Neural Networks. *Deep Learning for Natural Language Processing*.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior*.

Pachet, F.; Roy, P.; and Barbieri, G. 2011. Finite-length Markov processes with constraints. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Ventura, D. 2016. Mere Generation: Essential Barometer or Dated Concept. *Proceedings of the Seventh International Conference on Computational Creativity, ICCC* (June):22–29.

Ventura, D. 2017. How to Build a CC System. In *Proceedings of the Eighth International Conference on Computational Creativity*, 253–260.