Small-Scale Systems and Computational Creativity

Nick Montfort and Natalia Fedorova
Program in Writing & Humanistic Studies
Massachusetts Institute of Technology
77 Massachusetts Ave, 14N-233
Cambridge, MA 02139
nickm@nickm.com phd.natali@gmail.com

Abstract

Creative computational systems have often been large-scale endeavors, based on elaborate models of creativity and sometimes featuring an accumulation of heuristics and numerous subsystems. An argument is presented for facilitating the exploration of creativity through small-scale systems, which can be more transparent, reusable, focused, and easily generalized across domains and languages. These systems retain the ability, however, to model important aspects of aesthetic and creative processes. Examples of extremely simple story generators are presented along with their implications for larger-scale systems. A case study focuses on a system that implements the simplest possible model of ellipsis.

Introduction

For a variety of institutional, intellectual, and other reasons, the typical computational system developed to model or produce creativity is a sizable one. Some of these systems, such as Harold Cohen's AARON, even become lifelong projects of their creators, continuing to accumulate rules and heuristics for decades.

There are certainly virtues to large-scale systems, which have revealed a great deal about formal models of creativity and creative computing. We present the argument that small-scale systems can also make contributions, serving to complement more extensive projects and to lead into them. Specifically, the argument is advanced that it makes sense to welcome such systems in new ways in conferences, in thesis work, and in the developing large-scale systems.

Rather than directly claiming that these small-scale systems are creative based on some formal definition, we argue that they *engage creativity* and are relevant to larger-scale systems that have been argued to be creative.

Small-Scale Systems that Engage Creativity

Many of the systems that will be discussed here are small – often limited to around 1 KB – and most were developed in a matter of hours or days. These are not systems built around a model of creativity; many of them, in fact, were not created with any particular research purpose in mind. However, each of these systems does explore one or more aspects of creativity relevant to its domain. These systems, without modeling creativity directly, nevertheless inquire about creativity. They also can focus larger-scale investigations of creativity that implement complete models.

The systems discussed here all use randomness within some framework of regularity. It can be creative to introduce randomness in a context where, individually or as a culture, regularity is the norm – and vice versa. But the connection between regular elements (a recurring vocabulary, a poetic form, etc.) and randomness (deployed in many different ways) is much more complex, as is the question of when randomness is a quick and easy substitute for a more sophisticated process and when it is the best method. While we believe that small-scale systems can be used to address issues of randomness and creativity, full discussion of this topic must be left for later.

Creative Text Generators of the 1950s and 1960s

By 1952, Christopher Strachey's innovative and certainly small-scale love letter generator was running on the Manchester Mark I and producing texts such as "YOU ARE MY EROTIC APPETITE: MY SWEET ENTHUSIASM. MY LOVE FONDLY WOOS YOUR CURIOUS TENDERNESS. YOU ARE MY WISTFUL SYMPATHY." The system runs today in emulation (Link 2007) and has been discussed recently as "the first experiment in digital literature" (Wardrip-Fruin 2011). Its purpose, it seems, was not to shine with brilliance but to parody the formulaic process of love-letter writing. By being a parody of a banal writing process, this small-scale system did serve as a model – a model of a *lack* of creativity – and demonstrated that computational processes could relate to human writing processes.

In 1959 Theo Lutz published on his small-scale system to generate stochastic texts based on Kafka's The Castle, pairs of "elementary sentences" with a logical connective. These include (in English translation from the German) "A CASTLE IS FREE AND EVERY FARMER IS FAR." and

"NO COUNT IS QUIET THEREFORE NOT EVERY CHURCH IS ANGRY." By drawing on a well-known author and transforming the text in a way that intensified his disquieting juxtapositions, Lutz created a system with a literary purpose. His system's operation, and its results, were consonant with Kafka's description of a formally valid social system in which the particular combinations were often meaningless.

In the United States, and in connection with the Fluxus movement, Alison Knowles and James Tenney published the 1968 chapbook *A House of Dust*. It consisted of 20 connected sheets of computer paper on which a poem generated by a Fortran program was printed, with each stanza of the same form. An example:

```
A HOUSE OF GLASS
   IN A DESERTED FACTORY
      USING ALL AVAILABLE LIGHTING
         INHABITED BY COLLECTORS OF ALL TYPES
```

This project showed that variations of a regular stanza could be interesting when lined up and read one after the other, and that a creative language generator could produce a reasonably lengthy work that is compelling and worth reading.

"about so many things" and "Arrested" from Electronic Flipbooks Nannette Wylde, 1998

These very simple systems for text generation were written in Macromedia Director. They merely place some strings that are selected uniformly at random into a simple template, the nature of which is self-evident. "Arrested" presents a sort of stanza that describes different situations in which people are arrested, and is, as Wylde describes it, "a play on preconceptions regarding social, ethnic, religious, and political affiliations." In the case of "about so many things," the template is simply "He" followed by a sentence completion and the "She" followed by a sentence completion, to produce text such as: "He likes chocolate / She thinks things should be different." The sentence completions range from being rather gender-neutral to being quite different when applied to people of different genders. For instance: "feels stressful," "is a good parent," "has a crush on the teacher," "is a firefighter."

The READ_ME file for "about so many things" explains: "the activities are drawn from the same pool of possibilities. Any line of text could be applied to either subject. In essence, the work explores the release of societal constraints regarding gender roles." Many sentences have different connotations when associated with people of different genders. By simply assigning sentences at random to be about either "He" or "She," "about so many things" produces interesting texts that provoke the reader to think about cultural preconceptions related to gender. The lesson for large-scale creative text generator is that determining the gender of a character, or transforming the gender of a character in an existing story, can be an important decision that is part of the creative process.

"The Two" and "Through the Park" Nick Montfort, 2008

"The Two" builds on the core conceit of "About so Many Things." It uses even less text and only a slightly more complex template, such that the original Python program fits into 1 KB and the JavaScript version is not much larger. (The 1 KB limit is inspired in part by the demoscene, but also by poetic compression. While limitations of this sort are useful in many ways, and do enforce certain types of simplicity, they do not guarantee algorithmic simplicity or clear and readable code.) Two people described by their roles are introduced in the first line of each generated stanza; pronouns introduced in the second line require that the reader assign specific genders to the people in those two roles; and a conclusion involving both of them is provided:

```
The indigent turns to the librarian.
She smacks him.
They pray together.
```

In this case, two characters are introduced, the first of which is stereotypically male in U.S. culture. The second is usually culturally assumed to be female. Then, "she" and "he" appear on the second line, suggesting an obvious but disturbing resolution of reference: The librarian smacking the indigent. Since the typical reader's assumptions about the behavior of librarians and indigents will not line up with this interpretation, the reader may be compelled to consider the other interpretation, that the indigent is female and the librarian male. In either case, this generated text (and many of the texts that are generated) will challenge the reader's assumptions and stereotypes. This and other small-scale systems by this developer have been described and compared to systems of other sizes (Montfort, 2012).

Another 1 KB Python program that has also been made available in a JavaScript version is "Through the Park." This system is an attempt to build a very simple model of ellipsis or elision, the omission of part of a story. A carefully constructed list of sentences is reduced by a fixed number (removing sentences at random but keeping their order) and the resulting shorter story is output. The method of ellipsis has no intelligence or creativity to it, but with carefully constructed sentences it can nevertheless be effective. "Through the Park" is the subject of a short case study in the next section.

"The Semi-Automatic Doodle Machine" from Microcodes Páll Thayer, 2010

This tiny program (at 756 characters, "a bit longer than most" in the Microcodes series, as Thayer notes) produces some simple instructions for that non-artistic but potentially creative drawing practice known as doodling. The program first prints "Use a pencil and a 210mm x 210mm sheet of paper. Start with your hand at the upper-left corner." and then prints some instruction such as "With pencil up, move 8mm to the right," printing a new one endlessly each time ENTER is pressed.

As a creative text generator, the program is curious because it generates instructions rather than a story or poem. Of course, the program is not framed as generating creative writing, but rather that non-artistic form of drawing known as a doodle. Seen as a generator of visual art, the program is rather hilarious. It uses a person as a sort of plotter, inverting the typical relationship between "user" and computer. With its tedious, precise instructions about how to do a task that has no external value, it might be a parody of creativity assistance software. It also highlights how computation can be applied at different stages of the creative process, questioning whether the entire pipeline of creative generation needs to be built for a system to be effective.

"Through the Park": A Case Study

The small-scale system "Through the Park" is about as simple as it can be while incorporating any computational elements at all. It provides a highly simplified model of an important narrative technique, however, a technique useful in full-scale story generation systems.

The Importance of Ellipsis

One way of understanding ellipsis is as one possible tempo at which a narrative may be related. In this view, it is the leaping over of one or more events in no time at all, which corresponds to telling the story at the fastest possible speed – an infinite speed (Prince 1982). The importance of this narrative technique has been articulated by narrative theorists, including Seymour Chatman: "Ellipsis is as old as *The Illiad*. But ... ellipsis of a particularly broad and abrupt sort is characteristic of modern narratives" (1978, p. 71). These omissions can allow the reader's imagination to fill the story in, as Fielding explains at the beginning of book III of *Tom Jones:*

> The reader will be pleased to remember, that … we gave him a hint of our intention to pass over several large periods of time ...
>
> In so doing, we do not only consult our own dignity and ease, but the good and advantage of the reader: for besides that by these means we prevent him from throwing away his time, in reading without either pleasure or emolument, we give him, at all such seasons, an opportunity of employing that wonderful sagacity, of which he is master, by filling up these vacant spaces of time with his own conjectures; for which purpose we have taken care to qualify him in the preceding pages.

Understanding ellipses has been the subject of some research, but generating ellipses has not been as well-studied. As recently as 2006, it appeared that computational narrative systems did not incorporate an ability to use ellipsis (Gervás et al.). Those in the field have noted the relevance of this technique to cinematic and textual story generation, however. The interactive fiction system Curveship (Montfort 2007 p. 107) can generate ellipses but does not determine how to do so. Ellipsis was also supported in the Mimesis system, because "narrative effects in [3D] environments are often achieved by selecting elements of the story world to elide from the narrative discourse (e.g., temporal and causal ellipsis) ..." (Young 2007 p. 14)

A Minimal Ellipsis System

"Through the Park" was prompted by a conversation with Michael Mateas about how to develop the simplest story generator grounded in a meaningful narrative technique. The first version of it, a 1 KB Python program, was posted on *Grand Text Auto* on November 20, 2008. It has 25 sentences. Nine are removed during execution and the remaining 16 are printed in their original order. The sentences are:

> The girl grins and grabs a granola bar.
> The girl puts on a slutty dress.
> The girl sets off through the park.
> A wolf whistle sounds.
> The girl turns to smile and wink.
> The muscular man paces the girl.
> Chatter and compliments cajole.
> The man makes a fist behind his back.
> A wildflower nods, tightly gripped.
> A snatch of song reminds the girl of her grandmother.
> The man and girl exchange a knowing glance.
> The two circle.
> Laughter booms.
> A giggle weaves through the air.
> The man's breathing quickens.
> A lamp above fails to come on.
> The man dashes, leaving pretense behind.
> Pigeons scatter.
> The girl runs.
> The man's there first.
> Things are forgotten in carelessness.
> The girl's bag lies open.
> Pairs of people relax after journeys and work.
> The park's green is gray.
> A patrol car's siren chirps.

The system is meant to tell a version of, or at least alludes to, the folktale Little Red Riding Hood. On *Grand Text Auto,* readers were asked if they considered a system this simple to be a story generator. While not all commenters agreed that it was one, game developer Gregory Weir was the first to reply, echoing Fielding in some ways:

> It's definitely a story generator. I like how my interpretation of the story can vary drastically on which cues are included. This is partly due to a few sharply-charged cues: the girl's smile, the knowing glance, the blank stare, and the police siren. Depending on which of these are included, cues like the girl's bag or the movement can be erotic or horrific.
>
> It does depend heavily on the mind's ability to fill in gaps … (Montfort 2008a)

The sentences were consciously written to suggest (although not directly assert) that the two characters might be in a friendlier or more antagonistic relationship, and that the situation is more playful or sinister.

Developing this generator led to an improved understanding of ellipsis and of the characteristics (both ontological and linguistic) of story elements and their representations. In this simple system, there is no representation of the underlying fabula or story levels that is separate from a potential text, which may or may not be included in the final, realized discourse. Linguistically, it is problematic to include pronouns or other words that refer to other sentences; if such words are used, "she" or "he" might appear before "the girl" or "the man" are introduced. The more cohesive a text is, the harder it is to elide a sentence from it without adjusting the other sentences.

The underlying events in a story also should be able to stand apart, but for narrative interest, it is appropriate that they are, in Weir's terms, "charged" with varying emotional implications. While it seems valuable for the events to be of different valences, it is also helpful that they contribute to a consistent scenario and agree on, for instance, who the two main characters are and what the setting is.

A more general model would allow different events/sentences to have different probabilities of being omitted; an even more general one would allow for conditional probabilities. Since experience with "Through the Park" suggests some qualities of the relationship between intersentential cohesion, the relationship between underlying events, and the opportunity for ellipsis, there are insights that could be applied in the development of more elaborate systems.

Generality across Languages

Gregory Rabassa has stated that "translation is essentially the closest reading one can give a text" (1989), suggesting that the translation of a computational system to produce linguistic or narrative creativity would at least have to involve a very deep analysis and understanding of the system. Large-scale systems are seldom translated because of the great effort that would be needed; small-scale systems are more manageable and can be translated in fairly short amounts of time, sometimes even by volunteers.

Fedorova translated "Through the Park" to Russian, demonstrating that the system does not only work in English. The small size of the system and simplicity of its operation facilitated this. The need to maintain an ambiguity of tone or emotion did complicate the translation process to some extent, further highlighting the particular way in which the original sentences were constructed. However, each of the sentences could be translated, resulting in a Russian system that produced ellipses with the same sorts of effects as the original English system.

Because "Through the Park" works at the sentence level, modifying the discourse without making adjustments to syntax, it is less language-specific than some other creative text generators are. "The Two" uses the ambiguity of gender of noun phrases in its first lines to achieve its effect; this ambiguity is not easy to achieve in all languages.

Generality across Story Domains

A system that is so specific that it can only tell one story, or one class of stories, is probably not worth much time or attention. While large systems are often difficult to convert to other story domains, adaptation is a good sign that the system is general. In the case of large-scale systems, such adaptations would often be difficult and time consuming; they are easier in smaller-scale systems.

The simple underlying system in "Through the Park" was re-used by writer and artist J. R. Carpenter to create two story generators, "Excerpts from the Chronicles of Pookie & JR" and "I've Died and Gone to Devon." The former program was used to produce much of the text of Carpenter's book *Generation[s]*. "Excerpts" was ported to JavaScript in 2009 by Ravi Rajakumar (independently of the port of "Through the Park"), translated into Spanish and Catalan in 2011 by Laura Borràs Castanyer, and translated into Russian by Natalia Fedorova in 2012.

Another system that uses "Through the Park" as a basis is Fedorova's "Halfway Through." This system has one Russian and one English array of sentences; it mingles an inner soliloquy with overheard phrases.

"Through the Park" is not the most-reused small-scale creative text generator (for instance, Montfort's *Taroko Gorge,* which was also originally a 1 KB Python program, has been appropriated and reworked online more than ten times) nor the most-translated (for instance, Montfort's "The Two," another originally 1 KB Python program ported to JavaScript, has been translated to French, Spanish, and Russian). Still, that it has been ported, translated, and re-used attests to its accessibility and flexibility.

Benchmarks, Baselines, and Subsystems for Larger-Scale Systems

A small-scale system can be used as a benchmark or baseline for evaluating larger-scale systems use similar techniques, driven by more elaborate methods. For instance, it could be worthwhile to compare a sophisticated system that elides parts of a story for a particular purpose (to generate suspense, to increase reader interest) against a system that elides at random, as "Through the Park" does. Even without a purpose-built story, such a system would reveal something about how effective the technique of elision or omission is when applied without any special logic, intelligence, or creativity. As a first step, developers of a creative system for ellipsis should show that it can exceed, by whatever metric, the effectiveness of a random one.

If an elaborate creative system to address one particular aspect of story-generation does not exceed the small-scale baseline, all is not lost. A larger-scale system that incorporates several subsystems can simply use the simple, random system to deal with that particular technique (ellipsis, assignment of gender to characters, or something else) while using more elaborate methods elsewhere.

Allowing for Small-Scale Work

Small-scale systems can be of direct as well as indirect significance. They can be easily understood and modified, even without the involvement of their original creators. The new systems that are developed in this way can contribute to new types of cultural production, having value inside and outside the computational creativity community. They can be provocative, challenging the ideas that have been developed using large-scale systems and helping to develop some that have been overlooked. They can be used in teaching as the starting point for literary work or more elaborate exercises in computational expression. Finally, they can be used to sketch, as an artist would, in preparation for undertaking a large-scale work.

Despite the worth of small-scale projects and the slight effort that is needed to execute them, the context of computer science, and many interdisciplinary contexts, discourages work on such sketches and encourages researchers to proceed more directly to the development of large-scale systems. There are a few cases where small-scale systems are seen to have a place – for use as examples, for instance, or as subsystems in a larger system – but not many.

A dissertation project usually corresponds to a large-scale system, and the master's thesis and undergraduate capstone projects are typically reduced versions. Most conference papers are based on work with large-scale systems; even short papers, such as this one, are often invited not for the discussion of small-scale systems but for the dissemination of intermediate results about work in progress.

Ph.D. students in every field are already expected to understand their research area thoroughly by reviewing and understanding the relevant literature. It seems appropriate for them to spend as much time as they would reading a handful of articles in the development of one, or a few, small-scale systems. Such systems allow for different perspectives and approaches to be attempted; they also encourage a focus on the essential and on extreme abstraction of method and of the domain of creativity.

There are institutions that support, or could support, the development of small-scale systems. In particular, the hackathon, codefest, demo party, or other sort of competition, as often arranged outside of an academic context as inside it, could be employed to encourage the development of small-scale creativity systems. Although adding such an event to an existing conference would not change the paradigm for system development radically – those who were able to attend and compete would be there because their paper about a large-scale system was accepted – an event for quick development of systems could call attention to the value of such systems.

Small-scale systems have definite benefits, despite the institutional preference for using and discussing large-scale ones. These systems are easily portable across platforms, easily translated, easily generalized to different domains, and capable of capturing the essential aspects of important narrative techniques. Since they are also quick to put together, it would be sensible to do more to allow and encourage their development.

Acknowledgements

Our thanks to the anonymous reviewers, particularly for the suggestions that led to the discussion of randomness and the section on benchmarks, baselines, and subsystems.

References

Fielding, H. 2011. *The History of Tom Jones, a Foundling.* In Wikisource. Modified April 7. http://en.wikisource.org/wiki/The_History_of_Tom_Jones,_a_Foundling

Gervás, P., B. Lönneker-Rodman, J. C. Meister, F. Peinado. 2006 Narrative Models: Narratology Meets Artificial Intelligence. In *Proc. of Toward Computational Models of Literary Analysis, International Conference on Language Resources and Evaluation.*

Knowles, A. and J. Tenney. 1968. *A House of Dust.* Cologne: Verlag Gebrüder König. Excerpted in Pearson, L, ed., It Is Almost That: A Collection of Image+Text Work by Women Artists & Writers, Los Angeles: Siglio. 2011. 194-199.

Link, D. n.d. *Manchester Mark I Emulator.* http://www.alpha60.de/research/muc/

Lutz, T. 1959. Stochastische texte. In *Augenblick* 4:1, 3-9. Trans. H. MacCormack, 2005. http://www.stuttgarter-schule.de/lutz_schule_en.htm

Montfort, N. 2007. Generating Narrative Variation in Interactive Fiction. Ph.D. diss., University of Pennsylvania, Dpt of Computer and Information Science. http://www.cis.upenn.edu/grad/documents/montfort_000.pdf

Montfort, N. 2008a. Story Generation in 1k. On *Grand Text Auto.* November 20. http://grandtextauto.org/2008/11/20/story-generation-in-1k/

Montfort, N. 2008b. The Two. On *nickm.com.* http://nickm.com/poems/the_two.html

Montfort, N. 2008c. Through the Park. On *nickm.com.* http://nickm.com/poems/through_the_park.html

Montfort, N. 2012. XS, S, M, L: Creative Text Generators of Different Scales. January. Technical Report, The Trope Tank. TROPE-12-02. http://trope-tank.mit.edu/TROPE-12-02.pdf

Prince, G. 1982. *Narratology: The Form and Functiong of Narrative.* New York: Mouton Publishers.

Rabassa, G. 1989. No Two Snowflakes Are Alike. In *The Craft of Translation.* Chicago: Univ. of Chicago Press.

Thayer, P. 2010. "The Semi-Automatic Doodle Machine" From Microcodes. January. http://pallit.lhi.is/microcodes/

Wardrip-Fruin, N. 2011. Digital media archaeology : interpreting computational processes. In E. Huhtamo and J. Parikka, eds., *Media archaeology: approaches, applications, and implications.* Berkeley, Calif.: Univ. of California Press.

Wylde, N. 1998. About So Many Things and Arrested. *Electronic Flipbooks.* CD-ROM.

Young, M. 2007. Story and discourse: A bipartite model of narrative generation in virtual worlds. In *Interaction Studies* 8:2, 177–208.