Generating Reports, Fiction, and Text That Sounds Good

NICK MONTFORT

Nick Montfort is a professor at MIT, is principal investigator in the University of Bergen's Center for Digital Narrative, and directs a lab/studio, The Trope Tank. He lives in New York City. His work includes ten computer-generated books (in print from seven presses) and a diversity of other digital projects. Montfort's latest poetry book, *All the Way for the Win,* is composed entirely of three-letter words. His six MIT Press books include *The Future* and two co-edited volumes, *The New Media Reader* and the *Output* anthology.

Abstract

Natural language generation (NLG) came into its own in the 1970s, using AI based on symbolic methods. Automated reporting is one thread of NLG work; another is research in storytelling and fiction generation. The former sort of systems "textualize" underlying data; the latter generate plots and/or produce a narrative discourse based on plots. I argue that these two types of research are very relevant to each other. Writing fiction often involves imagining an underlying textual actual world in which characters undertake actions, then narrating what happens in this world, just as automated reporters narrate based on real-world data. In a literary sense, today's large language models (LLMs) are very different from both automated reporters and storytelling systems. They let language play out from probability distributions over sequences of words. To understand non-LLM approaches, I survey both automated reporters (which have presented news about weather, seismology, sports, finance, and elections) and storytelling systems (which have narrated invented events, giving us insight into narrative and cognition). These two sorts of systems, and LLMs, have important differences, but can also inform each other.

Keywords: Natural Language Generation (NLG), AI, Large Language Models (LLMs), Reporting, Fiction

DOI XXX – URL: https://platform.openjournals.nl/CLWap (openjournals.nl)
Uitgever/Publisher: Vlaamse Vereniging voor Algemene en Vergelijkende Literatuurwetenschap (VAL)

Generating Reports, Fiction, and Text That Sounds Good

NICK MONTFORT

Introduction

When GPT-2 was first dramatically announced in early 2019, its corporate author said that the full version of the model would be too dangerous to release immediately because of its potential to generate fake news and that smaller versions would be released in stages (OpenAI 2019). In November, OpenAI relented and released the full model (Vincent 2019); now, many much larger and more capable large language models (LLMs) are available. Despite this organization's concerns (or attention-getting marketing), GPT-2 and other LLMs are actually not news-generation systems at all, fake or real. LLMs may be combined with other computer capabilities and might be used to intentionally deceive or inform in various ways, but the LLM itself, however impressive, is a probability distribution over sequences of words. A model of this sort answers the question: Given a particular text, what's an appropriate-sounding way to continue that text? This makes an LLM less of a fiction writer and more of a system for the production of textual glossolalia (speaking in tongues) or certain types of avant-garde poetry, for instance, the sort the Surrealists sought to write by surfacing the unconscious (Montfort 2024).

The way LLMs function is particularly evident when using a "raw" or pure LLM, without an "instruct" layer that results in a chatbot. The pure LLM is not trained using reinforcement learning from human feedback (RLHF), a technique used to try to make their results less offensive or to have them align with human interests in other ways. OpenAI, Google, Meta, and other companies have Generative AI systems now that may incorporate all sorts of simple or complex augmentations beyond the basic LLM and indeed beyond RLHF. Perhaps models include a list of "very bad words" that will not be generated under any circumstances? Perhaps they recognize arithmetic expressions and, rather than generating a textual continuation in an attempt to solve the arithmetic problem, simply send those problems to a calculator unit, solving them in the way a computer usually would? Perhaps they generate computer programs and run those to produce answers? These are just a few of the simplest ways an LLM can be elaborated into a Generative AI

system by a company. Academic researchers and those at companies who publish at conferences do of course give us ideas about how the foundational LLM have been elaborated, but, when it comes to the systems in most widespread use, these systems are closed and proprietary. We only happen to know of a few such modifications that have been made to the underlying LLM used in the current ChatGPT, for instance. Indeed, OpenAI has not published a peer-reviewed paper about this system even documenting the foundational LLM and how it was trained. Note also that these systems are always in flux, which makes it harder to speak about them, and means that anyone intending to experiment in a scientific way with them will be flummoxed.

So, let's restrict our discussion to the LLM itself, and to free/libre/open LLMs, as these are available and we know how these systems work and what data was used to train them.[1]

To use an LLM, after it has been trained and, if desired, fine-tuned, one provides a text and sets various parameters. One of these, for instance, is "temperature", which causes generated textual continuations to be more conservative and typical (if the value is low) or more innovative, unusual, and incoherent (if the value is high). Because these models are trained on large amounts of text, and have parameters that control text generation, is it suitable to call them text models. Strictly speaking, they are not really even *language* models, because speech and gesture are also part of language, and these models don't comprehend them. But we can be generous and agree that "large language model" is a reasonable thing to call these software systems.

Once we note that these are models of text itself, we can be impressed by the extraordinary way that they (and particularly recent, large LLMs) are able to produce cohesive texts. At the same time, we can understand that these models do not do any news-writing or indeed fiction-writing. They are not reporters who perceive events in the world and communicate them. They are not novelists or storytellers, either. Developing narrative fiction involves imagining an underlying textual actual world in which characters undertake actions, then narrating what happens in this world. LLMs are just very good at generating words that sound like they should follow the previous words. They simply produce text—uncannily and amazingly, to be sure. They do this, however, without "knowing" about or having a representation of any real or fictional world, and without modeling narrative (Montfort and Pérez y Pérez 2023).

Some of the misunderstanding likely arises from text sequences that are almost inevitably continued in ways we recognize as narrative fiction or as journalistic. "Once upon a time, a princess" is a traditional story introduction, so it happens that an LLM, very good at producing a

[1] For example, researchers created free/libre/open source models based on documentation of GPT-2; these include GPT-J and GPT-Neo. The BigScience initiative, a large-scale collaboration, developed BLOOM, trained on multilingual data and with 176 billion parameters. Companies have also released pure LLMs; an example is the 7 billion parameter model from French AI startup Mistral.

plausible continuation of texts, will almost always tell some sort of story to continue this text, usually an amazingly cohesive one. One completion by Mistral 7B, for instance, begins "… was born. She had hair of gold and the bluest eyes you've ever seen. Her father and mother were poor but happy people and they loved their daughter to no end". This definitely sounds like the beginning of a story, but there's something obviously strange about it: The princess is born to poor parents.

Along these lines, "WASHINGTON, D.C.—The White House announced on Friday morning" seems certain to be the beginning of a news report and will also be continued in a way that would have seemed amazingly fluent a decade ago. An example from the same LLM, Mistral 7B: "… that it was 'deeply saddened' by the death of a beloved 27-year-old family member. 'We are deeply saddened to hear about the loss of Cookie Monster', spokeswoman Sarah Huckabee Sanders said in a statement". The reader is left to determine what is odd about this text, and whether it seems like real or fake news.[2]

An LLM does not fundamentally "know" anything about princesses, Washington, D.C., the White House, Fridays, or mornings, in the sense of having some internal model of these. It has an immense storehouse of text completions and conditional probabilities to go with them, which can be applied to continue any text. This is an incredible ability, because for decades, as researchers had progressed in several different ways, there was no way to generate cohesive, diverse, human-like texts.

Previous work on text generation did make significant advances of many sorts, however. There are two different but related text generation traditions that, on the one hand, involve reporting *real news,* and, on the other, enact *storytelling* by modeling fictional worlds. Systems have long been developed to transform quantitative and structured data to natural language, presenting weather reports, seismological reports, and news about sports, finance, and even elections. Some of these rule-based systems show elaborate rhetorical capabilities. Some have seen significant and widespread use. To describe what these systems do, I use the term "textualization". Just as one can visualize data or even sonify it and listen to it, one can also textualize it and read it, and we can assess a system as a better or worse textualizer depending upon how it facilities our understanding of the data.

While certain story generators have used grammars to generate texts (embodying rules for narrative, but without any explicit world model), there is also a long tradition of such systems that model an underlying fictional world. These systems do not report on anything factual; they narrate (or report on) fictional events that the system itself invents.

[2] Both completions were generated using Ollama's mistral:text model ID 495ae085225b from August 2025.

By viewing text generation in a way that encompasses its history, it's possible to understand the connection between reporting systems and storytelling systems. This also allows us to properly situate the contributions of the LLM, and to note that the developers of systems for textualizing data, systems for inventing and narrating stories, and systems for continuing texts parametrically (LLMs) can be integrated together. Those working to develop them can learn from work in each area.

Almost all of the following examples of reporting systems and storytelling systems are documented and briefly discussed in the recent book *Output: An Anthology of Computer-Generated Text, 1953–2023,* which I edited with Lillian-Yvonne Bertram (Bertram and Montfort 2024). In particular, see the Reporting and Storytelling sections; output from PAULINE can be found in the Rhetoric, Oratory, and Lectures section.

Real News Generation from the 1970s to the 2020s

A very early system that reported on game play was PROTEUS by Anthony Davey. His 1974 dissertation (Davey 1974) describes how the system generates a textual narration based on data about how one particular game of naughts and crosses (aka tic-tac-toe) had been played. A report on one game begins: "The game started with my taking a corner, and you took an adjacent one. I threatened you by taking the middle of the edge opposite that and adjacent to the one which I had just taken but you blocked it and threatened me. I blocked your diagonal and forked you."

Over the years a fairly standard architecture for symbolic text generation was developed that included three stages: A document planner determined the high-level structure of the output; a microplanner made more fine-grained, but still abstract, decisions; and a realizer generated the final sequence of characters (Reiter and Dale 2000). This architecture or one similar to it has been used in many projects over the years.

One of these was a 2009 system called StatsMonkey that would report on baseball games and, importantly, could do so out of chronological order—rearranging the events in the narrative discourse is common in all sorts of news writing (Allen at al. 2010). What follows is the beginning of a story it generated, covering a Little League game:

> UNIVERSITY PARK—An outstanding effort by Willie Argo carried the Illini to an 11-5 victory over the Nittany Lions on Saturday at Medlar Field.
> Argo blasted two home runs for Illinois. He went 3-4 in the game with five RBIs and two runs scored.

> Illini starter Will Strack struggled, allowing five runs in six innings, but the bullpen allowed only no runs and the offense banged out 17 hits to pick up the slack and secure the victory for the Illini.
>
> The Illini turned the game into a rout with four in the ninth inning.

While there's a bit of disfluency ("allowing only no runs"), the system reports on the game in a way that would be meaningful to children and parents.

Significant work was also done on textualizing databases to make their contents easier to understand. Kathleen McKeown's TEXT (McKeown 1985) could describe entities in the Office of Naval Research database, including ships and ordnance. It could also make comparisons, describing the difference between an ocean escort and a cruiser in great detail, based on the properties of each. TEXT used schemata for these different purposes to determine the high-level structure of output at the document planning stage.

RAREAS, later commercialized as FOG, is a system to generate textual weather reports based on maritime weather data. It was developed for use in Canada (Kittredge, Polguere, and Goldberg 1986) and elaborated so that it would textualize in both English and French (Bourbeau et al. 1990). Since 2012, a system called QuakeBot has been used to produce automated reports about minor earthquakes in the Los Angeles area based on data from the US Geological Survey (Oremus 2014).

To conclude this section, I'll discuss two remarkable reporting efforts, one from the 1980s and one from the end of the 2010s. The former is PAULINE, a system to not only report on events, but to customize its rhetoric according to defined parameters. The latter was a project by Arria NLG to provide complete coverage of all UK elections for the first time—done entirely without LLMs and, indeed, without any type of machine learning.

Eduard Hovy's PAULINE (Hovy 1988) not only textualized; it embodied explicit rhetorical goals used in generation. In addition to generating shorter and longer reports that sounded like news stories of different sorts, it was able to produce a text beginning "I am angry about Yale's actions. The University had officials destroy a shantytown called Winnie Mandela City on Beinecke Plaza at 5:30 AM on April 14. A lot of concerned students built it in early April" and another beginning "It pisses me off that a few shiftless students were out to make trouble on Beinecke Plaza one day: They built a shantytown, Winnie Mandela City, because they wanted Yale University to pull their money out of companies with business in South Africa." Importantly, all the texts it generated were based on the same underlying data, and while the level of detail varied and some information was omitted in some texts, none of them were "fake". The texts presented different

viewpoints, used different registers of speech (from vernacular to professional), and sought to persuade in different ways.

The Arria NLG project resulted in English and Welsh news stories (Reiter 2019). Although it was done in late 2019, after the development of GPT-2, the project eschewed all machine learning approaches and used a traditional three-stage pipeline for text generation. There were a few rationales for this approach: There was no highly relevant training data, because no one had ever reported on elections for individual constituencies before. The BBC also wanted to ensure that its style would be used. Finally, the reporting needed to actually match the election results. Even at the dawn of the 2020s, because the need for accuracy and authority was definite, system developers chose to *textualize* using a symbolic system rather than produce plausible text completions.

Automated Storytelling from the 1960s to the 2020s

Keeping this research on reporting in mind, let's turn to developments in automated storytelling. Researchers who developed storytelling systems typically didn't pay much attention to the surface generation of texts. They initially focused on inventing stories where characters' actions were cognitively motivated. Then, there was more work done on developing interesting plots. Some researchers (myself included) have also considered how underlying events and existents can be effectively presented in a narrative discourse. Because of the focus of these systems, however, the language that is generated often sounds more formulaic and stilted than in reporting and journalistic systems. This is not because storytelling researchers don't care about language, the particular way that literary art actually manifests itself. It's simply because their research has dealt primarily with other aspects of the story composition process.

There are a few early systems that engaged with story and were interesting in their particular contexts. One of the earliest was a sentence generator by Victor H. Yngve, developed as part of a machine translation project (Yngve 1961). Although Yngve was not concerned with story or narrative per se, he chose to develop a grammar based on the first ten sentences of a children's story book. The sentences he selected are actually descriptive, or represent habitual action. So, in isolation, they are poor examples of narrative content. The grammatical, nonsensical outputs he generated were certainly quite striking, however:

1. When Engineer Small keeps Small and four fireboxes, he keeps driving wheels, his steam, it and four black and oiled fireboxes.
2. He has four polished sand domes.

3. He has four proud, little, polished, polished and proud boilers under proud bells, steam and the whistles in four whistles.
4. When steam is proud of the four fireboxes and four engines, the train is shiny.
5. When Engineer Small is proud, he has Small under a little and proud smokestack.
6. When he is proud and oiled, Engineer Small is polished.
7. Water is big.
8. When he is oiled, the shiny smokestack is proud of four engines.
9. A headlight is heated.
10. When he is heated, Engineer Small is polished.

In 1963 Joseph Grimes, a linguist working in Mexico City, developed a program to produce very short stories that were based on structures from Vladimir Propp's *Morphology of the Folktale*. He told generated stories to people who spoke indigenous languages in order to learn about how easily they would understand them, presumably with certain variations introduced (n.a. 1963). Later that decade, in the influential "Cybernetic Serendipity: The Computer and the Arts" (Reichardt 1968), one selection of computer-generated texts were "Little Grey Rabbit Stories". Eric Mendoza produced these plotless descriptions of animals in an Edenic environment, which did, at least, have some consistent style to them. And in the early 1970s, Sheldon Klein undertook an elaborate project to write detective stories (Klein et al. 1973). He called it a novel writer, although it produced only a few pages of repetitive narrations of game playing, infidelities, and the like before a murder would occur and be solved. The participants in all the actions were determined at random.

These very early systems used a sentence grammar, templates, or simple statistical techniques to produce story-like surface language, without developing a model of a story world that could be narrated. It was in the mid-1970s that research work began on simulating deeper aspects of narrative. James Meehan did significant early work by developing TALE-SPIN, a system that modeled characters, their goals, and characters' thinking about other characters and their goals (Meehan 1976). It could tell stories about animals who needed food (and needed to ask each other for help) or humans who were interested in sex. To generate actions, the system gave characters goals and then used a technique known as planning to find a way they could accomplish them, basing each character's sequence of actions on everything that character knew and felt about other characters.

The second half of a TALE-SPIN story about George Ant and Wilma Bird is as follows:

George was very thirsty. George wanted to get near some water. George walked from

> his patch of ground across the meadow through the valley to a river bank. George fell into the water. George wanted to get near the valley. George couldn't get near the valley. George wanted to get near the meadow. George couldn't get near the meadow. Wilma wanted George to get near the meadow. Wilma grabbed George with her claw. Wilma took George from the river through the valley to the meadow. George was devoted to Wilma. George owed everything to Wilma. Wilma let go of George. George fell to the meadow. The end.

Although there seems to be no mechanism to produce this effect, the story may seem to be ambiguous about George's fate. TALE-SPIN does have an entity (a character) representing gravity. Did falling to the meadow allow George to quench his thirst, or did it lead to his demise? There's nothing that explicitly describes Wilma's feelings for George, after all. My reading, given that George is an ant, is that the fall didn't harm him, that he and Wilma get along, and that Wilma was helping—but that's open to interpretation. Meehan himself presented several "mis-spin tales" that were generated before bugs in the system had been worked out. Those discussing TALE-SPIN have often found these to be as interesting as the final, successful stories.

This intricate system was in many ways a milestone in story generation. It also suffered from what Noah Wardrip-Fruin called "the TALE-SPIN effect" (Wardrip-Fruin 2009). Wardrip-Fruin identifies this with reference to "the ELIZA effect", a response that people had to Joseph Weizenbaum's mid-1960s chatbot, ELIZA/DOCTOR, which used an extremely simple technique to simulate a conversation with a Rogerian psychotherapist. The ELIZA effect is the way this simple system led people to be extremely impressed, to overread, and to believe in the intelligence and even humanity of the system—even as the rules behind ELIZA and the DOCTOR script were simple and could be described in a short paper. The TALE-SPIN effect, on the other hand, involves a system that has tremendous, powerful conceptual and cognitive modeling under the hood. Despite all of this work the system is doing, it's unimpressive, because it doesn't expose that intricate model in an interesting way.

I will relate a few high points of the history of narrative fiction generation, organizing the discussion thematically rather than in a strict chronology. A decade and a half after TALE-SPIN, the researchers who developed a much simpler system, TAILOR, used a similar planning technique, and also generated animal stories (Smith and Witten 1991). Their idea was that as characters pursued goals, they could arrange these so that conflicts arose, leading to interesting stories. An example:

> Once upon a time there was an arctic tern named Truman. Truman was homeless. Truman needed a nest. He flew to the shore. Truman looked for some twigs. Truman found no twigs. He flew to the tundra. He met a polar bear named Horace. Truman asked Horace where there were some twigs. Horace concealed the twigs. Horace told Truman there were some twigs on the iceberg. Truman flew to the iceberg. He looked for some twigs. He found no twigs. Horace walked to the shore. He swam to the iceberg. Horace looked for some meat. He found some meat. He ate Truman. Truman died.

Perhaps the use of deception makes this more interesting than some of the stories from TALE-SPIN? At the same time, it's not clear why Horace hid the twigs and then extensively pursued Truman, swimming to an iceberg. According to the description of the system's workings, Horace could have simply eaten Truman when they first met, making for a more efficient meal but a less interesting narrative.

Lyn Pemberton took a different approach from world modeling and planning in developing a system called GESTER to produce plot summaries in the style of Old French epics, based on her studies of the Chanson de Geste cycle (Pemberton 1989). Here is an example of what the system produced—four short paragraphs from a longer story:

> Charles and Aymeri broke into Narbonne.
> As a result of seeing Blanchefor Charles wanted Blancheflor.
> Charles succeeded in getting Narbonne.
> Charles praised God. Charles forgot to reward Aymeri. Charles threw Thibaut into prison.

GESTER did none of the modeling of action and cognition that was essential to TALE-SPIN. Instead, stories were generated by a grammar, analogous to a linguistic grammar for sentences of a language. The rules of the grammar, although syntactical, captured the meaning of aspects of the story world. Following the logic of Old French epics, the grammar does not allow Saracen men to pursue Christian women. On the other hand, a Christian knight can desire and pursue any Saracen woman, even one who is married.

Although the story grammar is a different approach than one based on representation of characters, their goals, and planning to accomplish them, there are ways to combine the two. In the mid-1990s, Imogen Casebourne's system Grandmother used both modes, with the framework of

the story being developed by a story grammar (Casebourne 1996). Within that generated story, action sequences—in the example provided, a character searching for an opponent in different places and finally encountering her and exacting vengeance—are developed using planning.

Scott Turner's MINSTREL was a highly influential system that used a technique called case-based reasoning and generated stories in response to formally represented morals (Turner 1994). It also consulted its own database of previously generated stories so as to avoid repetition. One of the simpler stories, generated in response to the moral "Pride goes before a fall", is:

> It was the Spring of 1089, and King Arthur returned to Camelot from elsewhere.
> A hermit named Bebe told Arthur that Bebe believed that if Arthur fought with the dragon then something bad would happen.
> Arthur was very proud. Because he was very proud, Arthur wanted to impress his subjects. Arthur wanted to be near a dragon. Arthur moved to a dragon. Arthur was near a dragon. The dragon was destroyed because Arthur fought with the dragon. The dragon was destroyed but Arthur was hurt. Arthur wanted to protect his health. Arthur wanted to be healed. Arthur hated himself. Arthur became a hermit.

Characters in MINSTREL's stories mistakenly perceive events, as when Lancelot sees the woman he loves kissing a man in another generated story. He slays the man, who turns out to be her brother. MINSTREL was compelling because of the model of creativity employed and the ability for characters to make new sorts of errors of perception and judgment (Ryan 2025). The stories were realized in a surface narrative style reminiscent of TALE-SPIN, with sentences that included "At the same time, Lancelot's horse moved Lancelot to the woods". As was typical in this sort of research, Turner's focus wasn't on the way the narrative was expressed.

MEXICA is a system, originally from 1997 and continually developed and used in research by Rafael Pérez y Pérez since then, which generates stories about the people who lived in the Valley of Mexico before the arrival of Europeans (Pérez y Pérez 1999). (To be more precise, the system is a plot generator, with textual outputs presented using a simple template-based method.) It models the dynamic emotional bonds between characters and the level of tension as each event transpires. The system is also meant to model the creative writing process, using a cyclical model of engagement (letting ideas flow freely) and reflection (being critical and revising). It therefore attempts to identify which of the events are more or less compelling. A short English example story from the first book publication of MEXICA's output (Pérez y Pérez 2017), which also presented each story in Spanish, follows:

That day the warrior's heart was full of energy.
The eagle knight was a proud native of the Great Tenochtitlán City.
A bad spirit took the warrior's soul, leading the warrior to get intensely jealous of the eagle knight.
The warrior threatened the eagle knight to kill him. He decided to hide in the Popocatépetl volcano.
This was enough! The eagle knight went to look for the warrior in order to confront him.
The eagle knight observed the warrior carefully. Then, he attacked him.
The warrior insulted the eagle knight because he was irritated with him.
In only a moment, the warrior and the eagle knight were punching furiously at each other.
Enraged, the eagle knight provoked and offended the warrior.
Striking quickly, the warrior injured the eagle knight.
The eagle knight treated his own injuries.
Quickly, the warrior ran away towards the Popocatépetl volcano.

While significant work was done throughout the 20th century in developing more and more interesting and well-motivated plots, there were still not a lot of effective ways to tell or narrate the underlying events of the story world. Given that the entire field of narrative theory deals to a large extent with the interplay between the story or content level and the level of narrative discourse, there seemed to be a great deal of work left to do. Of course, the *telling* of underlying events was exactly what non-fiction reporting systems were focused on, so this aspect hadn't been entirely neglected.

In fictional storytelling, a significant advance at the level of the narrative discourse was Mark Riedl's FABULIST, which innovated in generating the underlying sequence of story events, or fabula, but also generated a narrative discourse and dealt with specifics of media representation (Riedl and Young 2010). A later project focused entirely on the narrative discourse, or level of expression, was my own Curveship, which uses a three-stage text generator (Montfort 2011). Pérez y Pérez has collaborated with me and others to join MEXICA's plot-generating abilities with Curveship's capabilities as a narrator (Montfort et al. 2013).

Not all of the work done by storytelling researchers who have worked with computing involved developing formalisms and representations. A sort of precursor to text generation with

neural nets and LLMs, in that this system used "big data," was Say Anything by Reid Swanson, who sought to generate stories via user interaction, specifically, textual exchange (Swanson 2008). This was a different sort of system than the ones previously discussed in two important ways. One was the interactivity: A person would type a narrative sentence, the system would reply with a next sentence, and the loop would continue to a conclusion. The other was that it had no model of a story world, grammar of story, or other relevant deep representations. It continued the storytelling process by using a large database of segmented narrative blog posts, simply searching for the best match and returning the sentence that followed it. Although not a storytelling system, the concept behind the chatbot Cleverbot (originally known as Jabberwacky) is similar. It uses previous conversations that it has recorded for its source of data (Carpenter n.d.).

Textualization, Computational Narrative Fiction, and LLMs

With a better view of textualizing systems, storytelling systems, and LLMs, it's possible to understand more about what they are designed to accomplish and why (if one takes a different perspective) they may seem to have shortcomings. I hope the survey of work in reporting systems and storytelling systems, considered along with a discussion of what LLMs fundamentally do, will allow literary scholars to find new ways to read the generated texts from these three sorts of systems. Researchers working in these areas who look to history can also see how to combine these three types of systems and learn from the work done along each thread of research.

For literary scholars, poets, and others who are used to reading texts, the challenge in reading both computational narrative fiction and LLM output is that one really needs to look deeper and read not only surface texts but underlying systems. The anthology *Output* goes beyond reporting and narrative fiction, including a total of 200 selections that are meant to offer readers in the humanities and arts a familiar way into the complex innovations of computer text generation. This discussion was intended to provide a gateway for readers in literary studies, who can pursue the citations there to read more about the workings of the underlying systems.

Acknowledgments

The example outputs in many cases also appear in *Output: An Anthology of Computer-Generated Text.* I edited this book with Lillian-Yvonne Bertram, and our short introduction to many of these systems also informed this current work. Bertram's contributions have of course been crucial to this article. This research was partially funded by the Research Council of Norway Centres of

Excellence project number 332643, the Center for Digital Narrative.

Bibliography

Allen, Nicholas D., John R. Templon, Patrick Summerhays McNally, Larry Birnbaum, and Kristian J. Hammond. 2010. "StatsMonkey: A data-driven sports narrative writer." *AAAI Fall Symposium: Computational Models of Narrative* 2. https://cdn.aaai.org/ocs/2305/2305-9761-1-PB.pdf

Bertram, Lillian-Yvonne, and Nick Montfort. 2024. *Output: An Anthology of Computer-Generated Text, 1953–2023*. MIT Press.

Bourbeau, Laurent, Denis Carcagno, Eli Goldberg, Richard Kittredge, and Alain Polguere. 1990. "Bilingual generation of weather forecasts in an operations environment." *COLING 1990: Papers Presented to the 13th International Conference on Computational Linguistics* 3: 318–20. https://aclanthology.org/C90-1021/.

Carpenter, Rollo. n.d. Cleverbot. https://cleverbot.com

Casebourne, Imogen. 1996. "The grandmother program: A hybrid system for automated story generation." *Proceedings of the Second International Symposium of Creativity and Cognition*: 146–55.

Davey, Anthony. 1974. "The formalisation of discourse production." PhD dissertation, University of Edinburgh. https://era.ed.ac.uk/bitstream/handle/1842/8112/Davey1975Phd_full.pdf.

Hovy, Eduard. 1988. *Generating Natural Language Under Pragmatic Constraints*. Lawrence Earlbaum Associates.

Kittredge, Richard, Alain Polguere, and Eli Goldberg. 1986. "Synthesizing weather forecasts from formatted data." *Coling 1986 The 11th International Conference on Computational Linguistics* 1: 563–65. https://aclanthology.org/C86-1132/

Klein, Sheldon, J. F. Aeschlimann, D. F. Balsiger, et al. 1973. "Automatic novel writing: A status report." Technical report #186, University of Wisconsin Madison Computer Science Department. July 1973. http://digital.library.wisc.edu/1793/57816

McKeown, Kathleen R. 1985. *Text Generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press.

Meehan, James. 1976. "The metanovel: Writing stories by computer." PhD thesis, Yale University.

Montfort, Nick and Rafael Pérez y Pérez. 2023. "Computational models for understanding narrative." *Revista de Comunicação e Linguagens* 58: 97–117. September 2023. http://nickm.com/articles/Montfort_Perez_y_Perez__Computational_Models_for_Understanding_Narrative.pdf

Montfort, Nick. 2011. "Curveship's Automatic Narrative Style." In *Proceedings of the 6th International Conference on the Foundations of Digital Games (FDG '11),* 2011, pp. 211-218. http://hdl.handle.net/1721.1/67645

———. 2024. "Automatism for digital text Surrealists," *electronic book review (ebr)*. May 5, 2024. https://electronicbookreview.com/essay/automatism-for-digital-text-surrealists/

N.a. 1963. "Exploring the fascinating world of language." *Business Machines* 46: 11–12.

OpenAI. 2019. "Better language models and their implications." Blog post on OpenAI.com, February 14, 2019. https://openai.com/index/better-language-models/

Oremus, Will. 2014. "The first news report on the L.A. earthquake was written by a robot," in *Slate,* March 17, 2014. https://slate.com/technology/2014/03/quakebot-los-angeles-times-robot-journalist-writes-article-on-la-earthquake.html

Pemberton, Lyn. 1989. "A modular approach to story generation." *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 217–24.

Pérez y Pérez, Rafael. 1999. "MEXICA: A computer model of creativity in writing." DPhil Dissertation, University of Sussex. http://www.rafaelperezyperez.com/documents/MEXICA_DPhil_RPyP.pdf

———. 2017. *MEXICA: 20 years–20 stories (20 años–20 historias).* Using Electricity series. Counterpath.

Reichardt, Jasia, ed. 1968. Studio International 25s, special issue, July, "Cybernetic Serendipity: The computer and the arts."

Reiter, Ehud, and Robert Dale. 2000. *Building Natural Language Generation Systems.* Cambridge University Press.

Reiter, Ehud. 2019. "Election results: Lessons from a real-world NLG system," on Ehud Reiter's Blog. December 23, 2019. https://ehudreiter.com/2019/12/23/election-results-lessons-from-a-real-world-nlg-system/

Riedl, Mark Owen, and Robert Michael Young. 2010. "Narrative planning: Balancing plot and character." *Journal of Artificial Intelligence Research* 39: 217–68.

Ryan, Marie-Laure. 2025. Invited talk at Computational Models of Narrative. Geneva, May 28–30, 2025.

Smith, Tony C., and Ian H. Witten. 1991. "A planning mechanism for generating story text." *Literary and Linguistic Computing* 6 (2): 119–26.

Swanson, Reid. 2008. "Say anything: A massively collaborative open domain story writing companion." *Interactive Storytelling. ICIDS 2008*. Lecture Notes in Computer Science, vol 5334. Springer. https://doi.org/10.1007/978-3-540-89454-4_5

Turner, Scott R. 1994 *The Creative Process: A computer model of storytelling and creativity.* Lawrence Erlbaum.

Wardrip-Fruin, Noah. 2009. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies.* Software Studies series. MIT Press.

Vincent, James. 2019. "OpenAI has published the text-generating ai it said was too dangerous to share." *The Verge,* November 7, 2019. https://www.theverge.com/2019/11/7/20953040/openai-text-generation-ai-gpt-2-full-model-release-1-5b-parameters

Yngve, Victor H. 1961. "Random generation of English sentences." *Proceedings of the International Conference on Machine Translation and Applied Language Analysis,* 66–80.