1 Introduction

Every morning, Simon walks down the street to his local café in Wolverhampton. He orders a medium black filter coffee, which he nurses as he flips through the *Express & Star*, an independent regional newspaper. First, the cover story. Then, the less pressing items. The newspaper's contributors are good at what they do, tending towards fair representation of issues and citing relevant supporting data. Today, a story entitled 'Majority of New Mothers in Wolverhampton Are Unmarried' catches Simon's eye. He reads the story's introduction: 'The latest figures reveal that 56.5 per cent of the 3,476 babies born across the area in 2016 have parents who were not married or in a civil partnership when the birth was registered. That's a slight increase on the previous year.' *This is a sensitive issue*, thinks Simon. *Well, it takes a special kind of journalist to consider such a subject so objectively*. Simon continues reading the article, which cites figures and statements courtesy of the Office for National Statistics. It is not until he reaches the end of the text that Simon reads the following statement: 'This article has been computer-generated by Urbs Media, crafting stories and harnessing automation to mass localise.'

Simon looks up from the paper, quickly setting it down onto the table in front of him. He does not know what to think about a computer-generated news article. *You read things to connect to other people's stories*, he ponders. *If a computer writes a story, you take away that human connection. But here, there's no criticism, there's no opinion. I suppose it's so factual that I don't really mind if this article is computer-generated or human-generated*. Simon still does not know how he feels about computer-generated texts. The only certainty: *Now I feel like I'm on the set of a sci-fi film*.

This anecdote is fictitious but the technology is most certainly not. The 'unmarried mothers' article (Anon., 2017) appeared on the *Express & Star*'s website on 29 November 2017, as countless more computer-generated articles appeared in other news outlets worldwide that same day. However, most of these articles (including those by Urbs Media) actually do not include any disclaimers about their production processes. 'Crafting stories and harnessing automation to mass localise' is a slogan that Urbs uses to describe its service on the company's website

(Urbs Media, n.d.), but it does not accompany any of their articles. Moreover, Simon's conflicted feelings have been paraphrased from a series of focus groups conducted to discern readers' emotional responses to the idea of computer-generated texts (Henrickson, 2019c). While some may regard a computer's ability to generate intelligible narratives as being contained within the realm of science fiction, the technology that enables a computer to generate cogent prose has been in development for more than half a century. Computer scientists have long been engaged with programming computers to generate texts that are indistinguishable from those written by humans. Now we have reached a point when there are systems generating texts that we may read as part of our daily routines, unaware of their being computer-generated. The production of data-driven sport, weather, election, and business intelligence reports has been assigned to computers capable of producing these texts at a rate incomparable to that of humans, and on personalised scales that could hardly be considered efficient uses of time for paid human labour. Yet when we read these texts, we assume that they are social products, the results of human thought and intention, rather than computer-generated.

This is natural language generation (NLG). NLG is the computational production of textual output in everyday human languages. NLG systems are increasingly prevalent in our modern digital climate, as we have seen the emergence of numerous companies that specialise in generating output intended for mass readerships and readerships of one alike. Yseop works with an international insurance company to provide personalised explanations of refusals or partial agreements of credit insurance allocations (Anon., 2019). Phrasee generates marketing copy aligned with its many partners' unique brand voices (Malm, 2020). Narrative Science has worked in partnership with Deloitte to generate client-friendly narrative reports related to such issues as budget optimisation, financial operations, and internal auditing (Krittman, Matthews, and Glascott, 2015). These are only a few examples of NLG's current applications as implemented by a burgeoning industry based around this technology. Anyone who reads data-driven news has likely encountered at least one computer-generated text, perhaps unknowingly. Computer-generated texts are both online and in print, and they are everywhere.

This Element offers an introduction to the social and literary implications of NLG from a humanities perspective. More specifically, it examines how computer-generated texts challenge conventional understandings of authorship and what it means to be a reader. Accentuating the humanities perspective employed, this Element refers to NLG as 'algorithmic authorship', save for in discussions of NLG's technical functionality and in quotations. It should be noted that 'algorithmic authorship' is not a term coined for this Element. Some (generally popular) articles have used the term, but it has largely gone unused by NLG developers. This lack of widespread use may reflect the current emphasis on system development over output reception, for the term draws attention to the nuances of authorship as it applies to human writers and NLG systems. The use of 'algorithmic authorship' therefore distinguishes this Element's sociological focus from a process-focused computer science perspective: NLG is the technology; algorithmic authorship is the concept. While NLG refers to the process of text production, 'algorithmic authorship' more clearly refers to the social players – and the relevant sociocultural circumstances – involved in this technology's applications. 'Algorithmic authorship' situates NLG systems within the complex history of authorship – a history driven by both human and non-human agents – that is explored here.

NLG systems produce a wide range of texts for pragmatic and aesthetic purposes. News items, image and video descriptions, and prose fiction are just some examples of NLG's current applications. This Element focuses primarily on English-language prose texts for human readers, including works that are expository (e.g. news and business reports) and aesthetic (e.g. stories). The computer-generated texts cited in this Element exist in static and linear reading forms. Interactive applications of NLG – for example, chatbots – warrant their own analyses, and are thus only referred to when contextually necessary. Likewise, those computer-generated texts produced for reading by machines rather than humans – for example, content generated to promote Web traffic through search engine optimisation – are outside of this Element's scope, but have already been investigated elsewhere. Further, I do not deny the importance of genre differences in literary criticism, or wish to discredit readers' genre-specific expectations; expository and aesthetic texts prompt different reading practices and interpretive processes, as do texts of different genres. However, the focus of this Element is not one of literary analysis, but of readers' responses to the very concept of NLG itself. Once the stage has been set by such general research, future studies are better positioned to conduct genre-specific analyses. For now, concentration on textual output actually detracts from the more fundamental issue of determining NLG's unique contributions to the modern textual landscape: a landscape permeated with varied modes of human–computer collaboration. Without recognition of these contributions, one cannot fully appreciate the roles of NLG systems as cultural artefacts, as reflections of contemporary social values and needs, and in turn as re-enforcers or interrogators of these values and needs.

Of course, an NLG system's output must at least somewhat conform to readers' expectations of literary convention for that output and system to be relevant. Complete narratives comprise a beginning, a middle, and an end; genre conventions are anticipated; every text has an author. Indeed, it is not unreasonable to assume that a text reflects human agency, and that a text – regardless of genre – is an effort to communicate a predetermined message. With this assumption, readers engage their interpretive faculties to assign authorial intention, developing a perceived contract with the author. I refer to this author–reader contract as 'the hermeneutic contract' (Henrickson, 2018a). The establishment of an author–reader contract is warranted given the underlying assumption that through language we articulate and legitimise lived experiences to ourselves and others. Our perceptions of the world, and others' perceptions of ourselves, are shaped and shared by the words at our disposal. It is through language that agency is exercised, and that potentialities are realised. The hermeneutic contract is therefore rooted in an expectation of agency informed by lived experience.

Computer-generated texts in their current state complicate the hermeneutic contract. The hermeneutic contract rests on two assumptions: that readers believe that authors want readers to be interested in their texts, and that authors want readers to understand their texts. Even in instances of the avant-garde, authors are still presumed to have produced their texts with some sort of communicative intention that justifies unintelligibility; a reader's willingness to accept such writing as intentionally incomprehensible is for that reader to fulfil the hermeneutic contract. Ultimately, the hermeneutic contract supposes intelligibility for hypothetical readers who are able to engage with the social and literary conventions employed by what Tereza Pavlíčková (2013) calls the 'imagined author', in a nod to Wayne Booth's (1961) 'implied author'. The imagined author matters, Pavlíčková explains in a later paper with Ranjana Das (Das and Pavlíčková, 2013: 393), because its presence contributes to a reader's sense of familiarity with the writing subject, in turn helping to determine the reader's trust in a text's source. But what happens when the human author appears to be removed, and agency and intention may not be readily identifiable? The author of a computer-generated text is often an obscured figure, an uncertain entanglement of human and computer. With an eye to cultural vacillation, this Element offers a unique interdisciplinary consideration of a technology – NLG – that has been under development for more than half a century, but which has not yet been subject to any substantial analysis from a humanities perspective. Drawing from book history, media studies, computer science, digital humanities, and the social sciences, this Element offers a peek into the kaleidoscopic state of the art.

Section 2 – 'Discovering Natural Language Generation' – presents a technical introduction to NLG, defining technical terms as they are used within this Element: algorithm, program, system, and machine learning. It reviews the functionalities of some historical and modern NLG systems, as well as the history of systematised writing more generally. This section provides the necessary context for determining whether readers' perceptions of computational capability and expectations of system functionality are justified. Section 3 – 'The Development of Authorship' – offers a theoretical examination into which aspects of modern social life algorithmic authorship manifest and mobilise, arguing that algorithmic authorship reflects a current tendency towards individualisation. Section 4 – 'Algorithmic Authorship and Agency' – suggests a semantic shift from considering NLG systems as tools for actualising human ideation to NLG systems as social agents – and authors – in themselves. Such a semantic shift permits a more reflective discussion about the transformative power of these systems' output, explicitly recognising their distinct social contributions. Having presented a framework for reading computer-generated texts, this Element concludes with recommendations for further studies.

2 Discovering Natural Language Generation

2.1 Introduction

The year is 1845. Spectators gather in Piccadilly's Egyptian Hall, eagerly anticipating a demonstration of John Clark's Eureka machine. With the mere pull of a lever, the Eureka generates a grammatically and metrically correct line of Latin dactylic hexameter. For the price of one shilling, visitors witness the mechanised spectacle of the wooden, bureau-like contraption moving its wooden staves, metal wires, and revolving drums to produce a line of Latin verse that appears in the machine's front window. While each line adheres to a predefined system of scansion and strict syntactic formula – adjective | noun | adverb | verb | noun | adjective – the Eureka can churn out an estimated 26 million permutations (Hall, 2007: 227). Eureka! Mechanised text production is in the spotlight.

Not everyone was excited. An item in the 'Miscellanea' section of *The Athenæum*'s June 1845 issue (Anon., 1845) notes that Clark had worked on the machine for thirteen years 'as it would seem from the mere sport of the thing, and in a spirit of indifference as to what might be its subsequent use. . . . I do not see its immediate utility.' Other contemporary critics too expressed scepticism: one reviewer (Anon., 1846: 133) deemed the machine a 'useless toy', while another (Nuttall, 1845: 140) asserted that it was 'little better than a mere puzzle, which any school-boy might perform by a simpler process.' Modern scholar Jason David Hall (2007: 228), however, has described the machine as 'much more than a show-place diversion: this kitsch device – at once the technological embodiment of and a parody of Victorian prosodic science [the study of verse and meter] – was a literally interactive discursive site, the focus of a popular prosodic discourse that existed alongside institutional debates.' Hall sees the Eureka as a meeting place between those of academe and members of the general public (who could spare the shilling required to enter the exhibition). Indeed, the machine was a 'material embodiment . . . of the institutional practice of Victorian prosody' that spoke to academics and amateurs alike (Hall, 2007: 234). According to Hall, the Eureka evoked curiosity in various publics at the onset of the golden age of automata: machines programmed to operate on their own, often imitating the actions of humans or animals. Mechanised writing seemed especially enticing, with some American and British newspapers reporting on (fictitious) writing machines that tended towards producing poetry (Anon., 1841a; Anon., 1841b; Anon., 1844).[1] The Eureka serves as an early tangible example of mechanised writing's longstanding appeal to mass curiosity.

Technically, the Eureka was primitive; it just randomly applied words in accordance with strict programmed constraints. 'All it really did,' Jason David Hall (2017: 130) explains, 'was combine a limited array of integers, uniting predictability (all values were determined beforehand) with unpredictability (the order in which the values might be arranged was not).' However, the enthusiasm garnered for the Eureka's mechanised production of verse speaks not just to the Victorians' fascination with automata, but also to a continued enchantment with technological capability. The Eureka seems more a mechanical toy than any convincing form of autonomy, a bibelot rather than a bard. Nevertheless, recent efforts to restore the machine (Eureka AHRC Project, n.d.) perpetuate the sense of curiosity about a technology that helped pave the way for what we now call NLG.

NLG enjoyed a surge of interest within the academic community throughout the 1970s to 1990s. This interest also permeated the popular sphere. In 1986, Arthur C. Clarke (2000) published a fictional short story about 'The Steam-Powered Word Processor', which Reverend Charles Cabbage uses to mindlessly produce his sermons. Even earlier, in 1954, Roald Dahl wrote about 'The Great Automatic Grammatisator': a novel-producing machine that one Adolph Knipe uses to render human writers obsolete. Despite such widespread interest, though, NLG research has only recently begun engendering distinct intellectual traditions, especially related to developmental approaches. As David McDonald (1986: 12) suggests, this may be a result of the individualised nature of systems. Indeed, few NLG systems comprising the field's lineage remain. Written in programming languages now extinct, or saved in digital formats that have deteriorated or disappeared altogether, many of these systems survive only through secondary literature. Further, developers tend not to build upon

[1] Thanks to James Ryan for his digital curation of these materials.

work that has already been done, perhaps for reasons of inadequate digital preservation and/or claims for intellectual property. The field is in disarray, with no comprehensive analysis of NLG output reception ever having been published. As a result, we do not know where computer-generated texts fit within our current conceptions of authorship and reading. This section offers the necessary technical context for succeeding sections, which examine where computer-generated texts fit within conventional understandings of authorship and what it means to be a reader as per the hermeneutic contract.

2.2 Technical Terminology

Preceding any substantial analysis of NLG, numerous technological terms must be defined. This is, as any etymologist might anticipate, not so easily done. Despite technological terms like ‘program’ and ‘algorithm’ seeping into common speech, the meanings of these words as characterised by historical and current usage are hardly stable or precise. The definitions provided here are working definitions for this Element and should not be considered comprehensive or static. Technological developments force alterations of all of these terms – alterations that are hardly consistent across fields of study. Rather than offer conclusive definitions, the following paragraphs simply offer semantic scrutiny that informs the following discussion.

Colloquial understandings of algorithms often synonymise them with computer programs. However, these terms are not synonymous. Using a programming language (e.g. Java or C++), a program executes a set of instructions provided by a developer to solve a problem through fulfilment of a predetermined task. Each of these instructions is an algorithm. A program may comprise just one algorithm, or it may comprise a series of algorithms. An algorithm is a function with at least one defined source of input that produces a defined output. In some cases, the defined output may be an instruction to refer to another algorithm. In all cases, algorithms must be unambiguously specific. This is demonstrated by an activity popular with Introduction to Computer Science teachers, during which students direct the teacher to make a peanut butter and jelly sandwich. When a student tells the teacher to ‘put the jelly on the bread’ (an instruction), the teacher may then dunk her hand into the jelly jar and messily slop the jelly onto the loaf.

The student must instead direct the teacher to 'grip the knife's handle with your right hand', and then to 'move your right hand, still gripping the knife, up approximately seven inches', and so on. Each of these instructions represents an algorithm. Recipe analogies are commonly used when clarifying what an algorithm is. One extra teaspoon of baking powder can collapse a cake.

This Element, however, refers to NLG *systems* rather than programs. This is because generated output more often results from a series of programs than from a single program. Each program informs the next program's functionality, and together these programs make a system.

But all aspects of a system may not remain static. There are, after all, numerous problems that cannot be solved with fixed input data, instead necessitating an ever-growing and/or unstable corpus. In these instances, machine learning algorithms are more suitable. Search engines employ learning algorithms to navigate the Web and provide more relevant results in a digital landscape constantly altered by the daily emergence of countless new websites. News platforms employ learning algorithms to summarise current events by drawing from myriad articles and social media posts. A machine learning system has by necessity achieved a sort of autopoiesis, capable of maintaining itself through a network of processes that ensures the system's continuous production, maintenance, and improvement. A machine learning system could be viewed as a small technological ecosystem, with limited autonomy within its specified domain. As its algorithms evolve in response to a sort of lived experience, the system becomes increasingly independent of its creator.

Given the semantic, grammatical, and syntactic complexities of natural language, inputting all of a language's rules (and exceptions) into an NLG system would be a daunting task for even the most skilled linguist. What is more, by the time all the rules had been inputted, the linguistic landscape would have changed. For these reasons, many new NLG systems employ unsupervised machine learning, which applies inductive logic associated with unlabelled data by using clustering and association techniques to detect patterns that humans might overlook. Rather than teach a system a language, a developer may instead teach the system *how to learn* language, thereby allowing it to update its vocabulary, grammar, and syntax according to actual contemporary usage. Tang et al. (2016: 2) identify two approaches to NLG: rule- or template-based approaches and machine learning approaches, with the latter now often employing recurrent neural networks (RNNs). Many historical systems have tended towards the former. A recent example of the latter approach comes from a team at Google Brain. In preliminary tests, the team's RNN used unsupervised machine learning to distinguish linguistic patterns across approximately 12,000 (mostly fiction) e-books, amounting to approximately 80 million sentences, from the English Google Books Corpora (Bowman et al., 2016). Instead of constructing sentences one word at a time through next-step predictions (e.g. through Markov chains, which use statistical probabilities to generate texts word by word), Google Brain's system can construct sentences in more complex ways that accommodate global concepts and 'big idea' conveyance, resulting in output that better mimics human writing styles. In another recent example, Microsoft's Twitterbot Tay (@TayandYou), 'a chatbot created for 18- to 24- year-olds in the U.S. for entertainment purposes,' likewise mimicked human writing styles (Lee, 2016). However, those writing styles – applied by Twitter users interacting with Tay – led to the bot's spouting lewd and racially charged remarks within hours. Tay was promptly deactivated and Microsoft issued a formal apology.

The Tay fiasco shows that, in any machine learning system, a developer must in some way specify what the system is intended to do, and must embed algorithmic preferences and constraints to ensure that the system aligns with its intended purpose. 'If the rules of legal chess are not built into a chess-playing machine as constraints, and if the machine is given the power to learn, it may change without notice from a chess-playing machine into a machine doing a totally different task,' Norbert Wiener warns in his 1950 *The Human Use of Human Beings* (205–6). 'On the other hand, a chess-playing machine with the rules built in as constraints may still be a learning machine as to tactics and policies.' Tay serves as just one modern affirmation of the need for constraints in machine learning NLG systems. With no moral bounds and limited programmed appreciation for the nuances of language, Tay was unable to avoid offence. Tay's computational capacity for creation was insufficiently curbed.