Bibliotekarstudentens nettleksikon om litteratur og medier

Av Helge Ridderstrøm (førsteamanuensis ved OsloMet – storbyuniversitetet)

Sist oppdatert 04.03.24

Om leksikonet: https://www.litteraturogmedieleksikon.no/gallery/om_leksikonet.pdf

Tekstgenerator

En mekanisme som skaper/genererer tekster. Vanligvis gjelder det en datamaskin som har installert spesiell programvare for formålet, som kan være å drive eksperimentell tekstproduksjon. Et sett av språklige regler kodes inn i et dataprogram slik at programmet med en viss grad av autonomi (selvstendighet) frambringer svært forskjellige tekster basert på språkreglene. Det er altså maskinell-kombinatorisk tekstproduksjon. Det etableres en slags abstrakt modell som så ved bruk av programvaren fylles med konkrete tekster (Balpe og Magné 1991 s. 27).

"The most commonly cited definition of "generative art" is from Philip Galanter: " 'Generative art' is an art practice where the artist creates a process, such as a set of natural language rules, a computer program, a machine, or other mechanism, which is then set into motion with some degree of autonomy […]" […] underlying the generation process are rules, decided by the originator and effected by the computer, or input from the user." (Sarah Cook i http://net.art-generator.com/publications_cook_en.html; lesedato 06.12.19)

Dette er tre lett forståelige setninger: "Bjørnen spiste fem sauer i bondens skog." "Spekkhoggere puster med lunger og er en delfinart som tilhører tannhvalene." "Petter slo Kari med en sykkelpumpe." De følgende er tre andre setninger som kan dannes med de samme ordene som i de tre setningene: "Kari spiste en sykkelpumpe og tilhører tannhvalene." "Spekkhoggere slo Petter og Kari og hver delfinart med en sykkelpumpe." "En skog med sauer spiste bjørnen." Hvis ikke kun ordene, men bokstavene fra de tre første setningene kan kombineres, kan det oppstå en enorm mengde setninger, f.eks. "Sykkelbondens delfinsau hogger spekk." Det skal ikke mange ord eller bokstaver til før det blir et uoverskuelig antall kombinasjons-muligheter. Slik fungerer også et vanlig alfabet: under 30 bokstaver kan brukes til å lage millioner av forskjellige ord.

Gjennom å bruke en datamaskin til å kombinere ord, setninger og verselinjer, kan tekstmengden raskt overskride det et menneske er i stand til å lese – teksten kan vokse inn i uendeligheten (Clément 2002). Hvis programvaren stadig endrer tekstmassen, kan ikke leseren være sikker på å kunne lese den samme teksten på nytt, og heller ikke på om andre har lest de samme tekstene (Jean-Pierre Balpe i

https://articlesdejpbalpe.blogspot.com/2013/07/regles-contraintes-programmes.html; lesedato 21.08.19).

Tekster produsert av tekstgeneratorer må leses på en annen måte enn tekster som er skapt langsomt av en vanlig skrivende forfatter med intensjoner. Den kolossale mengden tekster som kan genereres i løpet av sekunder, gjør ferdiglesing umulig, og gjør lesingen til en slags meta-lesing, altså lesing av fenomenet lesing. Tekstene har ikke verdi i seg selv, men er uuttømmelige, midlertidige og fungerer som et slags vitner om litteraturens uendelige, produktive muligheter. På denne måten overskrider de begrensningene og reglene i tekstgeneratoren (Jean-Pierre Balpe i https://articlesdejpbalpe.blogspot. com/2013/07/regles-contraintes-programmes.html; lesedato 21.08.19).

"The goal for the author of a combinatory work is not to produce the *best* literary expression of an idea, but the most interesting *range of possibilities* the literary system can produce." (Rettberg 2019 s. 43)

Algoritmer (logisk-matematiske regler) brukes til å skape nye tekster, i en uendelig tekstproduksjon (Clément 2002). Teksten "selvmultipliserer seg" via elektrisitet og programvare. Tekstgenerering engasjerer ikke bare programmerere, men også lingvister, psykologer og forskere på kunstig intelligens (Jacques Anis i https://www.jstor.org/stable/i40079126; lesedato 14.02.19) Målet er ofte å lage setninger som er syntaktisk og semantisk akseptable, dvs. på et visst nivå forståelige (Archibald, Audet m.fl. 2011 s. 70).

Dataprogrammet kan være kodet til å produsere semantisk og syntaktisk relativt forståelige setninger. Men de algoritme-skapte tekstene frambringes ikke for å bli "litteratur", dvs. noe estetisk formfullendt eller universelt gyldig. De er flyktige, forgjengelige representasjoner. Det skapes en "variasjonens litteratur" der det performative er hovedsaken (Balpe og Magné 1991 s. 20). Tekstgeneratorer bidrar til å problematisere hva tekster og litteratur er, og hvilken rolle forfatteren og leseren spiller (Archibald, Audet m.fl. 2011). Tekster blir "avhumanisert", altså løsrevet fra en forfatters unike stemme og autentiske opplevelse. Forfatteren er ikke lenger en nødvendig instans, heller ikke menneskets inspirasjon eller intensjon. Tekstene skapes snarere gjennom "ingeniørarbeid". Tekstgeneratoren skal gi mening gjennom en maksimering av semantikkens uendelige muligheter, men også gjennom avkontekstualisering av det verbale uttrykket. Tekstgeneratorers utopi (eller dystopi) er en tekstlig "uavgrenset avgrunn", en uendelig tekst eller bok (Clément 2002).

"One of the most frequent criticisms of poetry and story generators is that they produce nonsensical or even unreadable output, and this may well be the case for many text-generation systems. This does not necessarily mean that they are lesser works of art – the author may be striving for some other effect than producing

compelling poetry or prose, such as exercising a particular constraint or mode of conceptual writing." (Rettberg 2019 s. 42)

"Forfatteren" av tekstene i tekstgeneratoren er en "meta-forfatter" som bryter forbindelsen mellom skrift og subjektivitet, og skaper stor avstand mellom disse instansene (Jean-Pierre Balpe i http://articlesdejpbalpe.blogspot.com/2013/06/meta-auteur.html; lesedato 06.09.19). Det følges et vi-prinsipp, ikke et jeg-prinsipp. Meta-forfatteren skaper en maskin der målet er at maskinen produserer (en slags) mening, ikke kaos. "What is important in such a situation is not the product itself but the process which leads to the product." (Jean-Pierre Balpe sitert fra http://nt2.uqam.ca/ fr/dossiers-thematiques/lart-generatif; lesedato 25.02.20)

"For the poet to be able to claim the poems created by bots as her own, she must accept that the technology is a part of the art making process and a part of the poet. It can be very hard to guess whether a poem is computer generated or made by real, human soul. […] It might just be Roland Barthes's dream come true; the author is finally dead and only the poem remains." (Monsen 2016)

Tekstgeneratorer skaper ofte en type tekster som viser fram sine kreative logikker, men der meningen er eller kan virke ugjennomtrengelig. Språket peker tilbake på seg selv. Vi blir mer oppmerksomme på mulighetene for variasjon, mangfold og forandringer i språket. Men tekstene bør på et eller annet vis være akseptable, dvs. ikke fullstendig kaotiske. Bak noen tekstgenerator-prosjekter ligger en drøm om en uendelig (eventuelt poetisk) produktivitet. Både det tilfeldige og algoritmer, uberegnelighet og matematikk, anarkisme og kontroll kobles (Clément 2002). Ut fra den spesifikke algoritmen som er brukt, kan tekstene som blir til, f.eks. tilhøre en bestemt stilretning eller sjanger. Datagenererte tekster kan framstå som pastisjer, dvs. etterligne gjenkjennelige tekster, ved at vokabular og språklig stil får tekstene til å ligne på tekster av kjente forfattere (Clément 2002). Eller tekstene kan oppfattes som parodier, altså latterliggjørende versjoner.

Algoritmer kan frambringe mange tilfeldigheter i bokstav- og/eller ord-sammensetninger. "Generative literature tries to be on the side of the effusive superficiality of show. It wants to reconcile the literary activity with that of play and game: to separate literature from the sphere of reverential and deadly seriousness in which the whole classical tradition locks it." (Jean-Pierre Balpe i http://www.dichtung-digital.de/2005/1/Balpe/; lesedato 21.08.19) Resultatet kan bli nonsenstekster, som er relativt meningsløse og eventuelt komiske.

"The text, no longer regarded as "literary", now has to annihilate all reverence because a generative text can always be substituted by another one. Hence it is not the singular display which is at the heart of generative literature but rather the movement, the series of ever-changing displays of text. The computer culture is close to spreading, to dispersion." (Jean-Pierre Balpe i http://www.dichtung-digital.de/2005/1/Balpe/; lesedato 21.08.19)

“Generative literature, defined as the production of continuously changing literary texts by means of a specific dictionary, some set of rules and the use of algorithms, is a very specific form of digital literature which is completely changing most of the concepts of classical literature. Texts being produced by a computer and not written by an author, require indeed a very special way of engrammation and, in consequence, also point to a specific way of reading particularly concerning all the aspects of the literary time. […] I call “generative literature” a literature where the texts are produced through a computer by means of a set of formal rules, the use of any kind of algorithm, specific dictionaries and eventually knowledge representations. That means a literature of which the author does not write the final texts but which only works at the level of the high rank components such as: conceptual models, knowledge rules, dictionary entries and rhetoric definitions. A text without an author generally seems to be out of question. Such a designation seems to describe an impossible literature because, despite the fact we generally assume that there is a very strong link between a text and its author, in this case the author is separated from the text. In generative literature, there certainly also is an author but one who has not really written the text which is being presented to a reader, his function is not the one we usually assign to an author. The difference is: this author is something like a meta-author trying to define what literature is for him and how his literary conception can be formally described. The tools of engrammation he uses are totally different. But at the end of the process there are also texts.” (Jean-Pierre Balpe i http://articlesdejpbalpe.blogspot.com/2013/06/principles-and-processes-of-generative.html; lesedato 24.10.19)

Briten Alan Turing, som fra 1930- til 50-tallet hadde stor betydning som utvikler av teorier om datamaskiner og med sitt arbeid innen informatikk, kryptografi og kunstig intelligens, planla å lage et dataprogram etter modell av surrealistenes “cadavres exquis” (“herlig kadaver”). Det er en slags lek der en tegning ble skapt ved at person etter person tegnet en del av en helhet uten å vite hva helheten var. Inspirert av Turings ideer lagde hans kollega Christopher Strachey i 1951 et program som kunne skrive kjærlighetsbrev. Dette regnes som verdens første tekstgenerator (Archibald, Audet m.fl. 2011 s. 70), “the first known example being Christopher Strachey’s 1952 love letter generator for the Manchester Mark I computer” (Noah Wardrup-Fruin i https://games.soe.ucsc.edu/sites/default/files/nwf-BC3-readingDigitalLiterature.pdf; lesedato 21.08.19). En annen pioner var den tyske informatikeren Theo Lutz, som i et tidsskrift i 1959 publiserte dikt kalt “stochastiske tekster” som var lagd av et dataprogram som brukte de hundre første ordene i Franz Kafkas roman *Slottet* (1926) (Clément 2002).

Den såkalte Turing-testen gjelder kunstig intelligens og går ut på å la et menneske som kommuniserer med en datamaskin tro at det kommuniserer med et annet menneske. Testen er bestått hvis en datamaskin kan kommunisere like effektivt og fleksibelt som et menneske. Franskmannen Jean-Pierre Balpe publiserte en rekke datagenererte tekster i noen franske tidsskrifter uten å forklare at de var skapt av et

dataprogram og dermed uten at en person skrev bokstavene. Noen ganger blandet Balpe disse tekstene med vanlige, menneskeskapte tekster (skrevet på tastatur). Poenget var at det skulle være umulig for leserne å skille mellom de to måtene tekstene var produsert på (Jean Clément i Archibald, Audet m.fl. 2011 s. 70-72).

Amerikaneren Robert Gaskins skapte en tekstgenerator som produserte haikuer (Balpe og Magné 1991 s. 19). "I wrote a program to generate haiku, which was embedded in the idle loop of a campus CDC6400 and became the most prolific poet up till that date, with a selection published in an anthology of computer poetry edited by Richard W. Bailey (*Computer Poems*, 1973)." (Ganskins i https://www.robertgaskins.com/; lesedato 25.02.20)

"[T]he new media artist D. Fox Harrell has created GRIOT, a computational narrative program named after West African storytellers that is designed to produce haibuns, or prose haikus. These short linguistic snapshots are generated from users' inputs that are then run through a combinatory algorithm" (Piper 2012 s. 138).

Amerikaneren Charles Hartman stod bak *Prose* (1996). "At first, computer poetry sounds like an oxymoron; computer poetry must be a simulation – "virtual" – and for that reason there can be no there, meaning-wise. Yet, Hartman and his own machines made of words, that is, his inventive poetry programs, produce some remarkably "fruitful linguistic material" […] A poet and ace programmer, Hartman has been "interested in the complicated boundary between what computers can do with language and what they can't." He believes that they "can do something worthwhile in the way of poetry" […] he doesn't want to delegate poetry to machines, only to use machines to stimulate our own thinking about language and meaning. […] Hartman's work has genuine philosophical implications, for he addresses the problematic of the arbitrary and the random, chance and necessity, and the uncanny sound frequencies underlying writing itself as represented by the frequencies of letters. As letter sequences lengthen, a computer mouths oracular sounding utterances chosen from letters scrambled from input texts: "On cigar. Light hand. That box fixed. Cup supposing/ white with the cup supposing white inside that" reads part of one Hartman/computer collaboration. Nonsense of course, but Delphic nonsense." (Alec Marsh i https://www.amazon.com/Virtual-Muse-Experiments-Computer-Wesleyan/dp/0819522392; lesedato 06.12.19)

Kunstneren John Morris publiserte i 1967 artikkelen "How To Write Poems With A Computer", der han blant annet skrev: "The computer must pay attention to rhythm and sound, and must somehow link texture with semantics to make each one complement the other – all without becoming obnoxiously evident in its task. It must grow banal when speaking of banalities, cool or crisp for the displeased mistress, hot and languid for a summer shower. At times it must play with the sheer sounds of words" (Whitman's 'Weapons shapely, naked, wan')". Morris programmerte tekster til å bli haikuer, skapt fra en liste med ord som programmet plukket tilfeldig fra, "on the fly". Tre av haikuene fra prosjektet ble slik:

“Frogling, listen, waters
Insatiable, listen,
The still, scarecrow dusk.”

“Listen: I dreamed, was slain.
Up, battles! Echo these dusk
Battles! Glittering …”

“Fleas spring far, scarecrow,
Oh scarecrow, scarecrow: well, far,
Scarecrow, oh scarecrow.”

“Many people generate poetry using computers, from artists exploring the effects of algorithms on language, to Internet hobbyists, to computer scientists interested in making artificial intelligence creative. Despite these varied authors, and lack of communication between communities, the techniques used to generate such poetry can be boiled down into a few simple categories with well-defined relationships. We define these categories as follows. In mere generation, a computer produces text based on a random or deterministic algorithm. All generative poetry systems we have come across use some form of mere generation. In the remaining two categories, the results of mere generation are modified and enhanced. This occurs either through interaction with a human (Human Enhancement), or through the use of optimization techniques and/or knowledge bases (Computer Enhancement). The results of mere generation can appear nonsensical, though this is not always a bad thing from an artistic perspective. By bringing in knowledge about words and the world, and by setting artistic goals, both human and computer enhancement drive generative poetry towards coherence and artistic style.” (Carolyn Lamb m.fl. i https://archive.bridgesmathart.org/2016/bridges2016-195.pdf; lesedato 25.02.20)

I 1960 gikk en gruppe franskmenn sammen om å danne et “Verksted for potensiell litteratur” (“Ouvroir de littérature potentielle”, forkortet Oulipo). Gruppa bestod av forfattere, matematikere og akademikere. En av forfatterne var den surrealistisk inspirerte Raymond Queneau. Hans “diktsamling” *Hundre tusen milliarder dikt* (1961) er en bok med sonetter, der hver eneste verselinje kunne blas om separat. Boka bestod altså av en stor mengde strimler med verselinjer, innbundet som en bok, og med milliader av kombinasjonsmuligheter. Leseren kunne gjennom å bla i strimlene lager hundre tusen milliader sonetter. Dette kombinasjonsprinsippet ble senere digitalisert av andre enn Queneau, for eksempel fantes det i 1999 en svensk versjon på Internett (på adressen http://www.ling.umu.se/~heldner/queneau2/dikter.htm). Andre har blitt inspirert av samme prinsipp: “Based on the Queneau model, a generation of 4,000 poems by [Georges] Perec, produced with drawings from Fabrizio Clerici which also obey the combinatorial rule.” (http://www.altx.com/ebr/ebr10/10sus.htm; lesedato 12.09.19)

“By inventing procedures for generating texts, the Oulipo separated the formal aspects of writing from its content so that procedures for making texts could be carried out independently of those who invent the procedures. […] When the Oulipo formed in 1960, one of the first things they discussed was using computers to read and write literature. They communicated regularly with Dmitri Starynkevitch, a computer programmer who helped develop the IBM SEA CAB 500 computer. The relatively small size and low cost of the SEA CAB 500 along with its high-level programming language PAF (Programmation Automatique des Formules) provided the Oulipo with a precursor to the personal computer […]. Starynkevitch used the machine to create an imaginary telephone directory composed of realistic names and numbers generated by his computer […] In 1981 the Oulipo published *Atlas de littérature potentielle* where they described some of the computer applications they had devised for reading literature. Their early experiments included machine-assisted readings of Queneau’s *Cent mille milliards de poèmes*. In this deceptively small book, Queneau had composed ten sonnets in such a way that the reader could select the first line of any sonnet, the second line of any sonnet, etc., and generate one of 10^{14} possible sonnets. The book itself contains the mechanism for generating poems: each line is printed on a strip of paper, and the reader can select strips from the original sonnets to generate a potential sonnet (Queneau 1961). Dimitri Starynkevitch had programmed his SEA CAB 500 machine to compose sonnets from Queneau’s *Cent mille milliards de poèmes*. In 1975 the Atelier de Recherches et Techniques Avancées, or ARTA, wrote a computer program that produced instantiations of the *Cent mille milliards de poèmes* as a function of a user’s name and the time it took him or her to type it. […] Paul Braffort and Jacques Roubaud, two Oulipians with backgrounds in mathematics and computer science, formed the Atelier de Littérature Assistée par la Mathématique et les Ordinateurs (ALAMO) in 1980 to explore computer-assisted writing. Following the model of Queneau’s *Cent mille milliards de poèmes*, the ALAMO wrote computer programs to produce texts according to the rules of various genres, such as poems and aphorisms.” (Mark Wolff i http://www.digital humanities.org/dhq/vol/001/1/000005/000005.html; lesedato 25.02.20)

“Braffort explained that combinatorial methods for generating texts with computers fall into two categories. The first category, applicational methods, involves templates for arranging words according to their grammatical function. One particularly amusing application generates what the ALAMO calls “Rimbaudelaires”, poems based on the structure of Rimbaud’s poem “Le Dormeur du Val” and composed of vocabulary from Baudelaire’s works […] Another example is Marcel Bénabou’s method for generating aphorisms (Bénabou 1980). Braffort developed a program that operationalized Bénabou’s algorithm by abstracting the structures common to adages and substituting new terms into the structures […] The potential of these computer programs resides in the way fragments of words and verses are recombined according to a set of well-defined rules. Poetic forms can thus be understood as algorithms for creating meaning with language. The ALAMO devised ways to formalize poetics in order for a computer

to generate structured texts which may or may not make sense. The actual poems produced by the programs are derivatives of the way computers can be harnessed to explore language. Reading these computer-generated texts can be amusing because of unexpected or incongruous combinations of words that oddly make sense. Despite their uncanny effects, however, texts produced through applicational methods still bear the mark of the inventor who not only determines the templates into which syntagma are inserted but also the stock of words and phrases from which the computer program draws." (Mark Wolff i http://www.digitalhumanities.org/dhq/vol/001/1/000005/000005.html; lesedato 25.02.20)

En hvilken som helst tekst "is only a temporary specimen of an infinite family of virtual texts. In concrete terms, this means that any point of the generative axis is the theoretical point of an infinity of texts […] What generative literature wants to affirm today is the vital and infinite power of the "literary" communication as a dynamic diffraction of relations, where the always-different text manifests its identities only through the infinite repetitions of its generation of the same, through its infinite changes more than through its halts. What this process assumes is the fecundating power of language as it enriches itself within all the restraining particularities of any given context. […] Generative literature wants first of all to be something like a "literarization" of technology, because what it demonstrates first and foremost in its multiplicities and its variations, are its potentials and its changing states. […] Generative literature's only pretension is to enrich the text's potentialities. It forsakes the fiction of fiction to be only interested in the subjective production and formalization of meaning. In that sense, it only exists through infinite literary production." (Jean-Pierre Balpe i http://www.dichtung-digital.de/2005/1/Balpe/; lesedato 21.08.19)

Østerrikeren Jörg Piringers "text-performance tool" *Nam Shub* (2006) "is a text processor, text generator and performance system. It is designed as a tool for both creators and performers of text and language oriented arts. […] functions to remove vowels or consonants, change the order of letters, split words into syllables, random operations on a word and letter level, complex substitution, text synthesis and tools for displaying text. Additionally all these functions can be combined and chained through a powerful scripting language and are therefore extensible. […] The discussed program and it's concepts are of course strongly influenced by the works and ideas of literary modernist avant-garde movements like Dadaism Surrealism, Lettrism, Oulipo, Wiener Gruppe and the Beat-poet's use of the Cut-Up technique. These movements and groups tried to extend the field of literature through the introduction of chance or in contrast through the implementation of strict rules for the generation of texts. […] Although *Nam Shub* is inspired by these early attempts it focuses on computer specific aspects of electronic poetry: dynamic and real time generation and manipulation of text." (Piringer 2006)

"William Chamberlain and Thomas Etter's *Racter* is one such notable system, as it resulted in what may have been "The First Book Ever Written by a Computer" as

claimed on the cover of *The Policeman's Beard Is Half-Constructed* (1984). In his introduction to the book, Chamberlain described the abilities of *Racter*. It "conjugates both regular and irregular verbs, prints the singular and the plural of both regular and irregular nouns, remembers the gender of nouns, and can assign variable status to randomly chosen 'things.' These things can be individual words, clause or sentence, forms, paragraph structures, indeed whole story forms." The texts published in *The Policeman's Beard Is Half-Constructed* tend toward the absurd, but they are largely coherent and more polished than many examples of generated text. Some commentators believed that the texts for the book were heavily edited. Regardless of the level of human input into the process or extent of editorial post-processing, the resulting texts often read as competent prose poetry, as in this example:

"A crow is a bird, an eagle is a bird, a dove is a bird. They all fly in the night and in the day. They fly when the sky is red and when the heaven is blue. They fly through the atmosphere. We cannot fly. We are not like a crow or an eagle or a dove. We are not birds. But we can dream about them. You can."

The book is also notable for its beautiful surrealist-style collage illustrations by Joan Hall, and it was also printed with a number of the poems angled diagonally or scattered across the page, reinforcing connections to historical literary avant-garde traditions." (Rettberg 2019 s. 39-40)

"*Mark V Shaney* was created by Bruce Ellis and Rob Pike in 1984 as a text generator for a fake Usenet personality. They implemented a basic Markov chain analyser and resynthesizer to generate text out of found texts. It was a very simple program without any possibility to control the results but through the access to a large corpus of text it could fool other Usenet users into thinking that those strange postings were produced by a real person. Markov chains as a text generating tool have since been widely used in text processing computer programs […] Andrew C. Bulhak follows a different approach with his *Dada Engine* and the *Postmodernism Generator*: the software generates random sentences by using recursive grammars. Depending on the structure and the encoded dictionary of the grammar it can produce for example nonsensical but grammatically correct philosophical essays […] Ray Kurzweil's *Cybernetic Poet* [1999] offers the user more control over the process of generating poems. It is in fact "disguised" as a poetic assistant that offers advice such on how to continue the text the user is writing in a text-editor-like interface. The program offers assistant "personalities" ranging from Blake to Yeats each processing their own literary corpus to suggest alliterating words, rhymes, possible next words or completing the rest of the line or even the rest of the poem. The actual generation process is hidden behind a recommendation interface so it is up to the user to follow the given advice or rather choose her own word or sentence." (Piringer 2006)

“Denne våren [2011] leverte Stian Hansen en hjemmeeksamen i filmvitenskap. Han hadde verken deltatt på forelesninger eller lest store deler av pensum. I kronikken skriver Hansen at han ved hjelp av fri fantasi, sitater fra It’s learning og The Postmodernism Generator, en nettside som genererer essay bestående av tilfeldige fagord, fikk karakteren C.” (https://www.adressa.no/student/article1681996.ece; lesedato 10.12.18)

“An even more playful project is C. P. Bryan’s *Cut ‘n’ Mix ULTRA* [2002]: it seems to be inspired by a mixing desk for audio signals. Four text tracks can be mixed together while controlling the “loudness” of each track. Additionally text effects can be applied such as randomly rearranging the resulting words, replacing words with synonyms, swapping words with randomly selected words of the same category and formatting the output to resemble song lyrics. The immediacy of changes in parameters and the real-time application of text manipulating functions are very similar to those in *Nam Shub* and follow the same idea of text generation as a kind of “poetry sculpting”. Taylor Berg’s *Darwin* [2005] enables the user to create poetry through a process that mimics genetic evolution. The user plays the role of natural selection by defining the fitness for survival through acting accordingly to his aesthetic preferences. Each time a user visits the project website he is presented six different automatically generated poems. After reading each poem he is asked to select the two he likes most and to enter their number into input fields. After pressing a button the algorithm generates a new set of six poems deduced from the two selected parent poems. Each new generation is recorded and can be reproduced and refined through the same selection process by each visiting user. This evolutionary mechanism can create very complex artistic results just by user driven selection of randomly generated structures. *Nam Shub* offers a similar feature for the algorithmic programming of new text-modifiying functions. Apart from Jean-Pierre Balpe’s elaborate text generators one particular project caught my interest as it used an aspect of human-computer-interaction that seems to be perfectly relevant for electronic poetry but is rarely used. *Labylogue* [2001] (by Jean-Pierre Balpe, Jean-Baptiste Barrière and Maurice Benayoun) was a networked interactive installation in the form of a virtual labyrinth built out of text walls. The users in three different cities could communicate by speaking into a microphone and move with a joystick through the text corridors. Simultaneously a computer equipped with a speech recognition software listened to the users’ voices and tried to understand what they were talking about to generate new texts for the walls accordingly.” (Piringer 2006)

“Nanette Wylde’s *Storyland* (2002), published in the *Electronic Literature Collection, Volume One* (2006), produces short stories, six-paragraph fictions that feature minimal interactions between three characters that read as minimalist contemporary parables. Consider one output:

“In the not-too-distant past, a misogynist cried for your sins. The misogynist was guilty.

Species dwindled.
The misogynist wrote a letter to a talk-show host. The talk-show host was also guilty when no one was looking.
While their inner storms were brewing, a bank teller lit a candle. The bank teller had a broken heart.
Money changed hands.
The bank teller was angered by the misogynist. The misogynist longed for the talk-show host."

While this story does not offer a lot of context, it does not come across as nonsense. This might well be kind of vignette or parable of contemporary sociopolitical life. It is not difficult to imagine that the misogynist is some kind of politician embroiled in a scandal, who goes on a talk show in order to perform a public *mea culpa*. The talk-show host meanwhile is just as corrupt as the politician. The bank teller, watching from afar, is disillusioned and heartbroken by the state of affairs, but nevertheless implicated in the same system that has produced the misogynist and the talk-show host, if powerless to change anything. This sort of story, like the output of many text generators, invites the reader's involvement not by providing an excess of detail but, instead, by providing the reader with a minimal sketch, with a great deal of interpretative space left for the reader to fill in. As readers we tend to have a desire to *make sense* of texts presented to us, minimal outlines such as this can serve as provocations, engaging our imaginations with prompts to flesh out a richer storyworld than actually denoted by the text that appears on the screen. […] Each time it is reiterated, the program pulls randomly from those arrays and sets the elements into place. For example, in the first sentence: [Time setting], a [stereotype character A] [past tense action verb] for [object]. The [stereotype character A] [past tense of a condition]." (Rettberg 2019 s. 41-42)

"Could a poetry generator produce poems of adequate quality to be published in literary magazines? This, roughly, is the challenge that Jim Carpenter set for himself in developing the *Erica T. Carter Project*, an ambitious poetry generator. The project used corpuses and styles (analyzed by Carpenter as "tree adjoining grammars") from famous poets such as Emily Dickinson, Frank O'Hara, Sylvia Plath, Gary Snyder, and Rachel Blau DuPlessis, and mixed their words and styles together algorithmically to produce new poems. In addition to developing a complex program, Carpenter pushed things a bit further, creating a virtual persona for the generator, and had her submit poems to literary magazines, a number of which were published. In 2004 Carpenter exhibited the generator and a collection of its output at the Slought Foundation in Philadelphia as "Erica T. Carter, The Collected Works" (Carpenter, 2004). In 2008, along with Stephen McLaughlin, Jim Carpenter released *Issue 1: Fall 2008*, a 3,785-page work that was allegedly a compilation of poems by more than 3,000 contemporary American poets. In reality, Carpenter's generator produced all of the poems. Carpenter's project was both a complex and accomplished poetry generator and a Dadaistic performance, which thumbed its nose at the poetry establishment. Many of the poets listed as authors

were in fact not pleased to find their names attributed to published poems that they had not written (Goldsmith, 2008).” (Rettberg 2019 s. 43)

“An example for an application that explicitly generates visual poetry is *Poem Generator* [2009] by Amorvita. It creates colourful constellations on the screen by either randomly choosing presets or by user supplied words and characters. The text is arranged randomly but obviously limited by constraints that give the result the appearance of concrete poetry like for example Eugen Gomringer’s work. […] Eugenio Tisselli’s *MIDIPoet* [1999] is a software that allows the manipulation of digital text and image in real-time.” (Piringer 2006)

Nick Montfort, Serge Bouchardon m.fl. lagde i 2008 verket *The Two*, som “could be called a digital poem or a story generator. It produces three-line narratives. In the first line of each stanza, two characters of unspecified gender are introduced. The second line includes two pronouns and a verb phrase, stating specific genders for the two characters but leaving the resolution of these pronouns up to the reader. The last line offers a sort of conclusion and describes something about the two characters. Because particular roles introduced in the first line (such as “the babysitter” and “the police officer”) are stereotypically imagined as mapping to particular genders, the story that is generated can pose a challenge to readers and can expose their assumptions. Because languages differ in how easy it is to initally omit mention of person’s gender, the translation of this piece can also be challenging.” (https://elmcip.net/creative-work/two; lesedato 06.12.19)

“From short stories to writing 50,000 word novels, machines are churning out words like never before. There are tons of examples available on the web where developers have used machine learning to write pieces of text, and the results range from the absurd to delightfully funny. Thanks to major advancements in the field of Natural Language Processing (NLP), machines are able to understand the context and spin up tales all by themselves. Examples of text generation include machines writing entire chapters of popular novels like Game of Thrones and Harry Potter, with varying degrees of success.” (Pranjal Srivastava i https://www.analyticsvid hya.com/blog/2018/03/text-generation-using-python-nlp/; lesedato 13.12.18)

Simon Biggs’ *The Great Wall of China* (1998) er basert på en ufullendt tekst av Franz Kafka kalt “Ved byggingen av den kinesiske mur”. Hele Kafkas tekst blir ved bruk av en tekstgenerator (som inneholder en kompleks syntaks-funksjon) omgjort til en enorm rekke av setnings-kombinasjoner. Nederst til venstre på skjermen dukker det opp nøkkelord for hva som styrer setningsmiksen i hvert tilfelle. I tillegg til massive tekster er det bilder på skjermen, og bildene skaper en romvirkning som får multiteksten til å framtre som en slags mur. Teksten forandrer seg på ulike måter hvis kursøren føres over denne “muren”, og man bare kan lese skikkelig hvis kursøren står stille (Christiane Heibach i Simanowski 2001 s. 35-36).

Et spesielt tekstgeneratorprinsipp følges i Noah Wardrip-Fruin med fleres programvare "The Impermanence Agent" (1999). Programvaren fungerer mens brukeren surfer på Verdensveven og samler ord og komponenter fra de besøkte nettsidene som så stables sammen til en fiktiv fortelling. Verbal tekst og bilder fra surfingen integreres til en "sammenhengende" historie. Det oppstår en "estetisk dokumentasjon" av brukerens søkespor i form av en helt personlig fortelling (Christiane Heibach i Simanowski 2001 s. 38). Det er tilfeldigheter som avgjør hvilke elementer fra de besøkte nettsidene som havner på bestemte steder i fortellingen.

"In 1948, British children's book author [Roald] Dahl published a short story called 'The Great Automatic Grammatizator' in which a machine writes such excellent fiction that its creator soon dominates the field of publishing. In 1950, the American novelist [Kurt] Vonnegut published a short story called 'EPICAC' featuring a fictional computer of the same name which wrote love poetry. […] British computer scientist Christopher Strachey's *Love Letter* generator [is] a variable text programmed on the Manchester University Computer in 1952. Wardrup-Fruin attributes to Strachey 'the first experiment with digital literature and digital art of any kind' (Wardrip-Fruin, 2011: 302). […] Vonnegut's fictional EPICAC computer reappeared in his novel *Player Piano* (1952), in the same year as Strachey's *Love Letter* generator. It is well within the realm of possibility that Strachey's enigmatic choice of the love letter as a literary form through which to test the random number facility of the Manchester University Computer was inspired by a work of print literature." (Carpenter 2017)

Epiphanies (2001) av Christophe Bruno er et eksempel på "Google Art". Brukeren skriver inn ord i et søkefelt, og søkemotoren Google finner deretter fragmenter av setninger fra en rekke dokumenter med disse ordene, som så av Brunos programvare blir satt sammen til en merkelig tekst. *Emotepoem* (2008) av Peter Howard har sju tematiske parametre som brukeren kan kontrollere, der graden av vold, erotikk, materialisme, skjønnhet, surrealisme, ro og intensitet kan justeres. For hvert parameter finnes det spesielle ord i en databank som tekstene genereres fra (Simon Brousseau i http://nt2.uqam.ca/fr/dossiers-thematiques/lart-generatif; lesedato 06.12.19).

En "oceanic (as opposed to terrestrial) nature of digital textuality has been nicely rendered in Nick Montfort and Stephanie Strickland's *Sea and Spar Between* (2010), which algorithmically combines the writings of Melville and Dickinson to produce as many stanzas as there are fish in the sea (about 225 trillion). As you pass over them with the cursor, the lines of verse writhe, wiggle, and shift like marine life." (Piper 2012 s. 164)

"Nick Montfort and Stephanie Strickland's *Sea and Spar Between* (2010) incorporates fragments from the sparse poems of Emily Dickinson (1831-1886) and dense prose from Herman Melville's novel *Moby Dick* (1851). The spaciousness of

Dickinson's dashes – 'you–too–' – merges with the oceanic churning of Melville's prose – 'leagueless sing and steep' – in stanzas assembled from words common to both and unique to each. These loosely coupled language systems create a vast verse-scape within the web browser window, chartable by longitude and latitude displayed at the bottom of the screen, and navigable by keystroke, mouse-click, or scroll wheel. Long-time collaborators, Montfort and Strickland interject human-readable critical commentary into their computer-readable source code, offering readers a number of ways into the text and inviting other authors to adapt and modify their work. Taking up this call, in 2013, Mark Sample adapted the source code of *Sea and Spar Between* to create a new work, *House of Leaves of Grass*, based on the combined corpus of Mark Z Danielewski's novel *House of Leaves* (2000) and Walt Whitman's poetry collection *Leaves of Grass* (1891-1892). The hybrid corpora of both these examples combine and thereby dissolve formal distinctions between works of poetry and prose. Both *Sea and Spar Between* and *House of Leaves of Grass* contain links to web pages which offer information on how to read the work. In keeping with their watery theme, Montfort and Strickland write: '*Sea and Spar Between* is a poetry generator, which defines a space of language populated by a number of stanzas comparable to the number of fish in the sea, around 225 trillion' (2010). In keeping with his house theme, Sample writes: 'The number of stanzas (stanza, from the Italian word for 'room') approximates the number of cells in the human body, around 100 trillion' (2013). Born of a process of reading and rereading a finite corpus of print literature, by dint of the volume of their potential output these variable texts court unreadability. Of *Sea and Spar Between* John Cayley asks: 'If we can only bring some minuscule portion of a huge virtual linguistic artifact into actual existence for our critical consideration ... does the work exist at all?' (Cayley, 2014: 17). These works exist as events, not artifacts. As such, they refuse close reading as a critical strategy." (Carpenter 2017)

Sea and Spar Between er et "generative" dikt og "a poetry generator which defines a space of language populated by a number of stanzas comparable to the number of fish in the sea, around 225 trillion. Each stanza is indicated by two coordinates, as with latitude and longitude. They range from 0 : 0 to 14992383 : 14992383. In the tradition of massive generative poems initiated by Raymond Queneau's *Cent Mille Milliards de Poémes*, this is an impossible text to read completely in a lifetime, requiring 6,421,232,876.71 years of reading, 24 hours a day, 7 days a week, 365 days a year (with a day of rest on leap years) – if you allot 30 seconds to read each stanza. Fortunately just as one doesn't need to navigate the seven seas to appreciate them, this poem doesn't need to be apprehended in its entirety to be enjoyed. And Montfort and Strickland have provided us with an interface that invites exploration in both serendipitous and precise ways." (http://iloveepoetry.com/?p=117; lesedato 06.05.15)

"Nick Montfort's *ppg256* poetry generators (2012) are a series of works that operate within an extreme constraint: the name of the project stands for "perl poetry generator 256 characters in length." Each of the poetry generators in the series is a

single line of code of exactly 256 characters. In this sense, *ppg256* is a form of conceptual writing. The author's goal is not so much to write a generator that produces rich imaginative writing or even poetry that would be published in a literary magazine but, instead, that can produce readable language in poetic forms: no small feat in itself, when the number of characters the programmer allows himself are fewer than those in this sentence. Montfort's project here is a *tour de force* in the computer science conception of "elegance": the idea that the best code is that which produces the most substantial desired effect, while utilizing the minimal computer memory and processing power necessary to do so. […] When run, it produces output such as this stanza:

the coat
bans no hack
moat no poat
mash of coed
moes at hams

Words and phrases are being produced here in an arranged format that resembles poetry. There is some rhyme, and some alliteration, recognizable pattern, but the result is just shy of language that suggests intended meaning. […] Montfort has in fact published a book, *#!* (2014a), which includes output from these minimal generators and others, along with the source code of the programs." (Rettberg 2019 s. 43-45)

Spine Sonnet (2011) av Jody Zellen er en diktgenerator som lager 14 linjers sonetter, "an automatic poem generator in the tradition of found poetry that randomly composes 14 line sonnets derived from an archive of over 2500 art and architectural theory and criticism book titles." (http://nt2.uqam.ca/fr/cahiers-virtuels/article/poetique-de-la-poesie-numerique-pour-ecrans-tactiles; lesedato 19.08.20)

Den danske *Tilfældigvis er skærmen blevet blæk* "er en poesigenerator laget som en fysisk installasjon på biblioteket i Roskilde. Opptil tre lesere kan delta samtidig, og hver leser tar tak i en lærinnbundet og sammenlimt bok som fungerer litt som en Wii-kontroll. Forfatteren Peter-Clement Woetmann har skrevet mange alternative verselinjer til diktet, og lesernes bevegelser og valg styrer hvilke linjer som blir del av "deres" versjon av diktet. […] det blir sinte linjer om de trykker hardt på boken og vennligere linjer om de har et mykere grep […] Når diktet er ferdig skrives det ut på en sånn smal lapp som bibliotekene pleier å bruke til kvitteringer, og diktene postes også automatisk til en egen blogg." (Jill Walker Rettberg i http://blogg.nrk.no/bok/2012/11/01/elektronisk-litteratur/; lesedato 11.01.18)

Ranjit Bhatnagars *Pentametron* (2012) "explains itself by clarifying that "With algorithms subtle and discrete / I seek iambic writings to retweet". The bot finds tweets unintentionally written in iambic (metrical "feet" with two syllables, where

every second syllable is stressed) and puts them together in an always-growing poem. This poem highlights the constantness and constant change in electronic literature mentioned earlier. The poem is never-ending; the technology generating the poem makes sure it persistently grows. The poem is constant and accessible in the sense that the reader can always just open their phone, look under their third arm, and enter the poem. The poem even sends "push" notifications reminding its reader to read the latest edition. At the same time, the poem is unceasingly changing as the bot updates itself. Every time the reader enters the page a new/same poem is there. Trying to find a section she previously enjoyed can prove almost impossible in the wealth of new material. This is what causes the poem to be both constant and interchanging. While some can say that this form of poetry makes for a lazy reader, because there is no work involved in accessing the poem, there is an acute need for attentiveness and appreciation. The poem might be gone forever the moment the hand returns to the pocket, buried under a mountain of new stanzas. […] The question quickly arises of *who* the poet really is when dealing with Twitterbots. Not because of the typical anonymizing style of the internet, but by the way the medium obscures the difference between human artist, Twitterbot poet, and poem. […] Is it art when the result is random? Where is the soul in something literally soulless? How can we measure quality when no human talent is involved in the actual making the poetry? Is the software programmer a creator of art in the creation of bots?" (Monsen 2016)

"Hva mangler når vi allerede har Google Bøker og Google Nyheter og Google Oversetter i tillegg til Google Søk? Google Poesi, selvsagt! En ny blogg ved navnet googlepoetics.com skaper dikt ut av de rare små setningene som dukker opp når man skriver inn et ufullstending spørsmål i den populære søkemotoren. En svensk leser skrev inn "En mi ..." og fikk følgende dikt fra sin *dator*: "En midsommarnattsdröm / en miljard / en miljon / en misstänkt liten kanelgiffel." " (*Morgenbladet* 12.–18. april 2013 s. 43)

I boka *Uncreative Writing: Managing Language in the Digital Age* (2011) undersøker Kenneth Goldsmith "a wide range of texts and techniques, including the use of Google searches to create poetry, the appropriation of courtroom testimony, and the possibility of robo-poetics […] Writers and artists such as Walter Benjamin, Gertrude Stein, James Joyce, and Andy Warhol embodied an ethos in which the construction or conception of a text was just as important as the resultant text itself. By extending this tradition into the digital realm, uncreative writing offers new ways of thinking about identity and the making of meaning." (http://cup.columbia.edu/book/uncreative-writing/9780231149907; lesedato 12.05.15)

En av den argentinske forfatteren Jorge Luis Borges' mest kjente noveller heter "Biblioteket i Babel" (1941) og beskriver prinsippene for en uendelig boksamling. Borges skildrer i detalj hvordan hans bibliotek er organisert, og forklarer at bøkene der inneholder alle tenkelige bokstavkombinasjoner, og dermed alle bøker som har vært skrevet og kan bli skrevet i framtiden. "The librarians search endlessly for any

book holding legible meaning, realizing there exists "no personal or world problem whose eloquent solution did not exist in some hexagon [med bokhyller]." While they at first rejoice in this realization, they soon become despondent with the hopelessness of finding anything they deem as meaningful within the endless shelves. They believe that "nonsense is normal" and "the reasonable (and even humble and pure coherence) is an almost miraculous exception." The narrator, however, disagrees, arguing that everything in the Library contains meaning, that every word or phrase or sentence, from "The Combed Thunderclap" to "Axaxaxas mlö" to "o iscfkln vwuhecmnv" can "no doubt be justified in a cryptographical or allegorical manner." It is impossible to create a combination of letters "which in one of its secret tongues do not contain a terrible meaning." This acts as a metaphor for the art of reading. Every text, be it a science textbook or a car manual or *The Library of Babel* can be interpreted by a reader as having meaning, or in fact multiple or infinite meanings." (https://hum11c.omeka.fas.harvard.edu/exhibits/show/open-readings/the-library-of-babel-and-infin; lesedato 03.09.20)

Amerikaneren Jonathan Basile ønsket å realisere Borges' imaginære bibliotek i form av et nettsted som rommer "bøker" med en uendelig mengde bokstav-kombinasjoner. Basile brukte programmeringsspråket C++, og ga åpningssida for nettstedet *Library of Babel* adressen http://libraryofbabel.info/index.html. Nettstedet genererer tekster i en ufattelig mengde, men det er nesten umulig å finne tekster i mengden som ligner på vanlige, leselige tekster. Meningsfulle tekster er forsvinnende få sammenlignet med alle tekstene som bare rommer tilfeldige bokstavkombinasjoner. Basile har altså lagd en tekstgenerator som eksemplifiserer og visualiserer Borges' uendelige boksamling. "The library creates a tantalizing promise of reason – somewhere in its pages are all the works lost in the burning of the Library of Alexandria, and every future masterpiece – but drowned out by infinite pages of nonsense." (Basile sitert fra https://www.flavorwire.com/515783/Brooklyn-author-recreates-borges-library-of-babel-as-infinite-website; lesedato 13.05.19)

I og med at det foregår en "økende entropi", indikerer det at kaos er det mest sannsynlige for ethvert system, at kaos er den mest naturlige tilstanden i universet (Archibald, Audet m.fl. 2011 s. 68).

Basile hevder at tekstene i hans *Library of Babel* er permanente, selv om de er generert av programvare: "Since I imagine the question will present itself in some visitors' minds (a certain amount of distrust of the virtual is inevitable) I'll head off any doubts: any text you find in any location of the library will be in the same place in perpetuity. We do not simply generate and store books as they are requested – in fact, the storage demands would make that impossible. Every possible permutation of letters is accessible at this very moment in one of the library's books, only awaiting its discovery. We encourage those who find strange concatenations [sammenføyninger] among the variations of letters to write about their discoveries in the forum, so future generations may benefit from their research. […] One guy

tried to find some meaning in the number of times he could find the names of various religious figures in the library. On the one hand, all words occur with a frequency which is based solely on their length, so 1 in every 20 million or so five-letter combinations will be jesus, while satan has the same distribution. On the other hand, who's to say these names and letters aren't significant in themselves?" (Basile sitert fra https://www.flavorwire.com/51578 3/brooklyn-author-recreates-borges-library-of-babel-as-infinite-website; lesedato 13.05.19)

"One user asks Basile if there are any issues with copyright or plagiarism violations, since an author's entire work exist in one form or another on the site, as well as work that haven't even been written yet. Basile claims the site most likely falls under fair use, since the work wasn't penned there and wasn't generated with intent for commercial use." (http://www.relativelyinteresting.com/the-labyrinthine-library-of-babel/; lesedato 08.09.20).

Litteraturliste (for hele leksikonet): https://www.litteraturogmedieleksikon.no/gallery/litteraturliste.pdf

Alle artiklene i leksikonet er tilgjengelig på https://www.litteraturogmedieleksikon.no