Half Constructed Infinity

On Algorithmic Literature and Text Generators

End Exam Thesis
Lydia Sachse, January 2012

Table of content

1) Introduction

    - Automation
    - The idea becomes a machine
    - Approach

2) The endless text

    - The first imaginations about writing machines
    - The computer – a creative author?
    - An objective thought

3) Automatic text generators

    3.1 How to make a Dadaist poem
    3.2 Stochastic Text
    3.3 Romance Writer
    3.4 RACTER
    3.5 ELIZA: “Does it please you to believe I am afraid of you?”
    3.6 BRUTUS

4) Conclusion

    - Simulated creativity
    - Combine
    - Explore
    - Transform

5) Timeline

Bibliogaphy

1
Introduction

"Carpets ... chairs ... shoes ... bricks ... crockery ... anything you like to mention – they're all made by machinery now. The quality may be inferior, but that doesn't matter. It's the cost of production that counts. And stories – well – they're just another product, like carpets and chairs, and no one cares how you produce them so long as you deliver the goods."

Roald Dahl, *The Great Automatic Grammatizator*

"When an artist uses a conceptual form of art, it means that all of the planning and decisions are made beforehand and the execution is a perfunctory affair. The idea becomes a machine that makes the art."

Sol LeWitt, *Paragraphs on Conceptual Art*

The subject of this text will be automation with special focus on text generators and algorithmic literature. Text generators are not limited to the computer. Already the invention of the movable type transformed religious and literary writings into algorithmic structures and even the systemic theory of rhetoric (Aristotle) was a step towards this direction. My research will focus on different examples of automatic text processors from the 20th century and how these emanate from each other in consideration of the technological background.

Inspired by mathematical thoughts scientists and artist started to experiment with computer generated text in the early fifties. Many writers got excited by the new possibilities of computer technology with the hope for finding new ways of artistic expressions. They thought formal constrains/regulations (given as instruction) could extend the expression of language. And the computer could function as a support for associative thinking. Indeed, the early experiments in algorithmic literature brought some surprising results. What started more like a literary game soon became a serious attempt to simulate natural language and to create well written, coherent stories automatically. With the help of the chosen examples, I will try to find an answer, if computer programs could be able to produce valuable text on its own. In favour of presenting a broad field of text generators I will not explain every concept in detail.

Automation

The concept of automation was coined by Ford Motor Co. Vice President Delmar S. Harder in 1948. In the manufacturing sense, it came from automatic + -tion.[1] The Greek term αὐτόματος means *automatos* – self-moving, self-acting, spontaneous. In the 20th century, the era of mechanization was replaced by the era of automation. Hence machinery became more than a hand tool which replaces muscular work. It seems that physical human work has

---

[1] *Online Etymology Dictionary*. Available from: www.etymonline.com

been replaced mostly by automated machines. These machines are capable of operating without external control or intervention. Its use became an inherent part of the daily routine. Therefore the need to control the dynamic behaviour of a system, became more and more important. Computer scientists, programmers, cyberneticians... etc., they all where studying the design and the function of the system for the purpose of making it more efficient by automation. But to automate human work, especially intellectual work, the system (the program) has to become very complex and divergent to consider as much cases as possible.

It's hard to imagine how language can be fragmented into 0 and 1 and put back together to create a literary masterpiece. To do so, one has to understand the mathematical concept of the machine – one has to "think" like a machine. And slowly the technological system transforms the human mind, to make it capable to install their ideas on the ever growing complexity of the machine.

The idea becomes a machine

To automate an idea, it has to become an instruction, a basic command with a logic- or at least understandable structure that can be executed by a person or a machine. An instruction can have different manifestations. It can leave the freedom of the decision to the performer or it limits him with restrictions and rules. It can be based on defined material and tools or it instructs the performer to choose for himself. In conceptual art the use of verbal instructions is quite common. „Draw a straight line and follow it“ (La Monte Young, Composition 1960) is a good example for a simple instruction that still allows any number of realisations because the tools are not defined neither is the final outcome. In comparison, the graphical, almost algorithmic instructions by concept artist Sol LeWitt are based on a predefined repertoire of shapes and rules (FIG. 1). LeWitt defined these rules for the design of a wall drawing (FIG. 2).

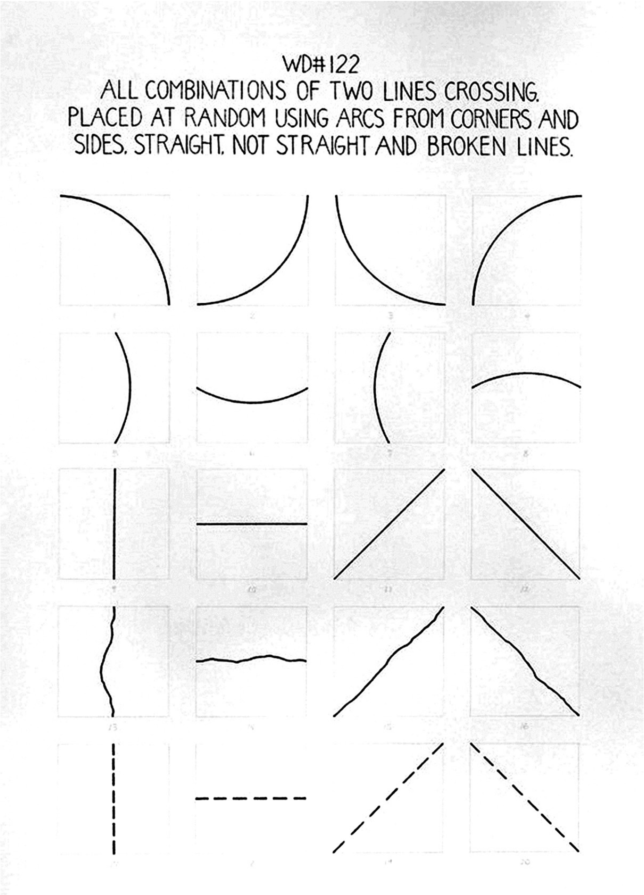
In informatics, we can find algorithmic[2] instructions that are performed by the computer. An interesting example, to illustrate the translation of a visual into an algorithmic instruction, is Casey Reas[3] implementation of Sol LeWitts wall drawings into the program (FIG. 3).

The like, algorithmic literature is not a creation out of nothing. It is based on material that is chosen from an existing predefined repertoire. This material (signs, words, sentences, texts) can be selected from a specialist field, a limited semantic field, from a story or the web... etc. Similar to the work of a conceptional artist the creator/programmer has to define a set of rules that he applies to the program. Rules can only be understood by the machine if they are formulated in machine language. To do so one really has to understand the structure of the machine and he must be

---

[2] *Algorithm:* pattern of action which describes how to achieve an aim in several steps (functions as work routine).

[3] Casey Reas is an artist whose conceptual and minimal works explore ideas through software. The instructions are expressed in different media including natural language, machine code, computer simulations, and static images.



WD#122
ALL COMBINATIONS OF TWO LINES CROSSING.
PLACED AT RANDOM USING ARCS FROM CORNERS AND
SIDES. STRAIGHT. NOT STRAIGHT AND BROKEN LINES.

FIG. 1 Instruction “Wall Drawing #122”

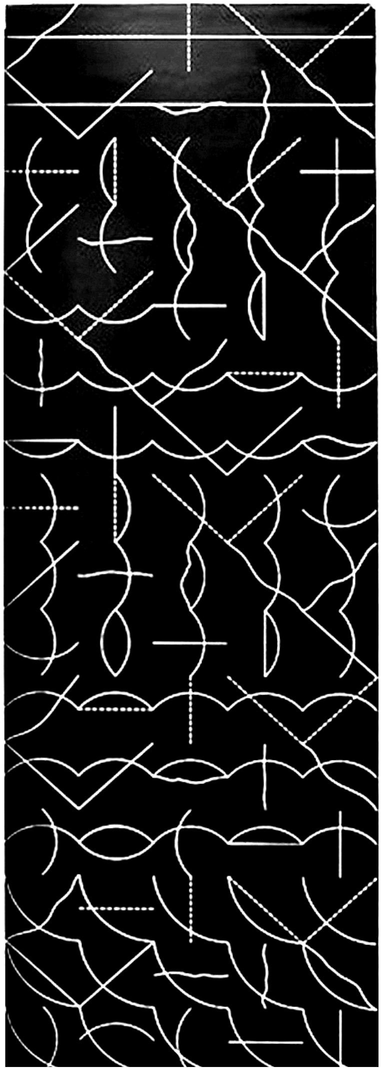


FIG. 2 Sol LeWitt, “Wall Drawing #260”

```
void setup()
{
  size(800, 600);
  noFill();
  stroke(255);
  background(0);
  framerate(12);
}
int res = 50;   // Resolution of grid
int arcres = 48; // Resolution of the arc

void draw()
{
  for(int i=0; i<height; i+=res) {
    for(int j=0; j<width; j+=res) {
      int r = int(random(4));
      if(r == 0) {
        arc_0(j, i, res);}
       else if (r == 1) {
        arc_1(j, i, res); }
      else if (r == 2) {
        arc_2(j, i, res); }
      else if (r == 3) {
        arc_3(j, i, res);
      }
    }
  }
}

 void arc_0(int xC, int yC, int r)
{
  int x = 0, y = r, u = 1, v = 2 * r - 1, E=0;
  while (x < y) {
    thin_point(xC + y, yC - x + res);
    x++; E += u; u += 2;
    if (v < 2 * E){
      y--; E -= v; v -= 2;
    }
    if (x > y) break;
    thin_point(xC + x, yC - y + res);}
}

 void arc_1(int xC, int yC, int r)
{
   int x = 0, y = r, u = 1, v = 2 * r - 1, E=0;
    while (x < y)
    { thin_point(xC + x, yC + y);
      x++; E += u; u += 2;
      if (v < 2 * E)
    { y--; E -= v; v -= 2;
     }
     if (x > y) break;
     thin_point(xC + y, yC + x); }
}

void arc_2(int xC, int yC, int r)
{
  int x = 0, y = r, u = 1, v = 2 * r - 1, E=0;
  while (x < y)
   { thin_point(xC - y + res, yC + x); x++;
     E += u; u += 2;
     if (v < 2 * E)
      { y--; E -= v; v -= 2;
      }
     if (x > y) break;
     thin_point(xC - x + res, yC + y);
   }
}

 void arc_3(int xC, int yC, int r)
{
  int x = 0, y = r, u = 1, v = 2 * r - 1, E=0;
  while (x < y)
   { thin_point(xC - x + res, yC - y + res);
     x++; E += u; u += 2;
     if (v < 2 * E)
      { y--; E -= v; v -= 2;
      }
     if (x > y) break;
     thin_point(xC - y + res, yC - x + res);
   }
}

 void thin_point(int x, int y)
{
  point(x, y);
}
```

FIG. 3 Program code represents Sol LeWitts *Wall drawings*

able to transform his idea into a mathematical step-by-step procedure. But there is no freedom in interpretation. While the instruction given by La Monte Young could be interpreted in many different ways, a computer knows only one way of reading the instruction. And algorithms do not consider exceptions! That is comprehensible, if we consider the architecture of the mechanical brain which can only understand on and off switches – 0 and 1. Once the instruction is implemented in the machine, the idea can be reproduced infinitely. That means an idea would never get lost because it would be stored in the memory[4] of a machine. It is even possible that the machine of the future could (re)write any book.

Approach

When dealing with this topic two popular questions could arise:
First: *Can the individual (the author) be imitated and finally replaced by a machine?*

---

[4] If we consider a hardware that can store data for an unlimited time.

Second: *Will the computer be able to produce a ground-breaking idea on his own?*

I try to give an answer to these questions in my conclusion.

2

The endless text

First imaginations about writing machines

We can find back fictional ideas about writing machines in different literary genres. In 1726 Jonathan Swift published his novel *Travels into Several Remote Nations of the World* (Gulliver's Travels). In Book III Gulliver visits the Academy of Lagado, the school of languages, where "useless projects" are undertaken. There, he is given a demonstration of a word machine (FIG. 4), which functions like a giant computing machine and is used for making sentences and books. The system consists of a frame and different cranks. The professor, who invented the system, "had emptied the whole vocabulary into his frame, and made the strictest computation of the general proportion there is in books between the numbers of particles, nouns and verbs"[5].

---

[5] Swift, Jonathan, pp. 44–45

This machine can render all kinds of text and even an unexperienced writer could generate a masterpiece with a bit of luck. “[...] and the professor showed me several large folio, already collected, of broken sentences, which he intended to piece together, and out of those rich materials, to give the world a complete body of arts and sciences”[6].

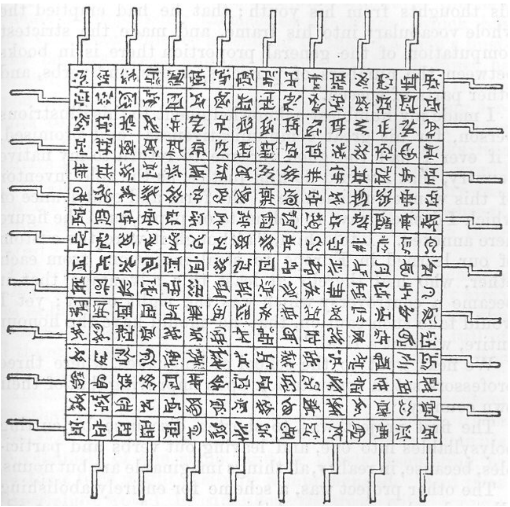


FIG. 4 Writing machine of Lagado

---

[6] Swift, Jonathan, pp. 44–45

228 years later, in 1954, the British author Roald Dahl, wrote a bizarre short story on literature production, called *The Great Automatic Grammatizator.* About a man called Adolf Knipe, who invented a machine that can write prize-winning short stories and novels within minutes. First, Knipe choose a wide variety of words from different literary fields and stored them in the memory of the machine. Than he developed a complex procedure to produce the desired story:

> “The genius of Adolph Knipe had [...] constructed a marvellous new control system which enabled the author to preselect literally any type of plot and any style of writing he desired. There were so many dials and levers on the thing, it looked like the instrument panel of some enormous aeroplane. First, by depressing one of a series of master buttons, the writer made his primary decision: historical, satirical, philosophical, political, romantic, erotic, humorous, or straight. Then, from the second row

> (the basic buttons), he chose his theme: army life, pioneer days, civil war, world war, racial problem, wild west, country life, childhood memories, seafaring, the sea bottom and many, many more. The third row of buttons gave a choice of literary style: classical, whimsical, racy, Hemingway, Faulkner, Joyce, feminine, etc. The fourth row was for characters, the fifth for wordage –and so on and so on–ten long rows of preselector buttons."[7]

To write the algorithm for the machine he started to analyse the grammar of magazine stories, that, in his opinion, where all based on similar, almost mathematical principles. He convinced his boss of the financial benefits of machine-generated stories. And with the argument: „There isn't any reason why we shouldn't put your name on some of the better stories“. He had no intention to tell anybody about the real source of the stories, that

---

[7] Dahl, Roald.

soon got published in every magazine of the country.

The three described machines may got build by different reasons and the authors probably had other intentions than to glorify automatic text production. But what these machines have in common, is the ability to produce infinite algorithmic text. What possibly inspired Dahl in his writing, is the idea, that one day automation will replace human labour. His story takes a very optimistic view on the process of machine-writing. But the capability to produce readable grammar of English is much more difficult and complex than it appears in Dahls story. That is why computer scientists took almost 60 years to develop natural-language generating programs, that could write grammatically correct stories. One could say, Roald Dahls approach is fictional and it would never be possible to fully automate the whole production of books.

But in 2007 Philip M. Parker, professor of management science, patented a method to automatically produce books from a template which is filled with text material and images from a database or the internet. The software translates original language into every alternative language, it creates titles, the cover, table of content, it makes advertising for the created book, and it writes the content. Every step is fully automated. Amazon lists Parker as the"author" of more than 200,000 different titles. Like Adolf Knipe, Parker does not indicate that the content is pieced together by a machine.

> Parker: "There is a need for an automated system that eliminates or substantially reduces the costs associated with human labour, such as authors, editors, graphic artists, data analysts, translators, distributors, and marketing personnel. Moreover, there is a need for an automated system that allows title materials to be quickly authored, marketed, and/or distributed as the latest possible edition of the material and in the format and written language of the end-user's choice."[8]

Parkers approach and Knipes intention have one thing in common: they want to "be able to produce the right stuff, at the right time, whenever we want it. It's a matter of business."[9] But there is also a difference, Parker uses undefined input material from the world wide web, hence the quality of the final book depends on what the algorithm finds.

The computer – a creative author?

For Michael Lebowitz, a computer science professor who researches and promotes artificial intelligence (AI) and automated text generation"the machine is not supposed to replace the author but to assist [...] and supports creativity by offering plot possibilities with which the author can work"[10]. Creativity... isn't it the keyword in

---

[8] Parker, Philip M. (2007). *Method and apparatus for automated authoring and marketing*. US patent.

[9] Dahl, Roald.

[10] Simanowski, Roberto."Text Machines". p. 103.

the question of the difference between computational and conventional literary production? If Lebowitz is right, all creative power comes from the author.

We can assume that an important part of the creative process is free association. But if the“incorporation of chance operations is only a weak simulation of the associative, intuitive and analogical components of inductive reasoning”[11], it should be possible to simulate free association with a computer program, or perhaps not? That leads us to the utopian idea of a universal machine that computes endless text and eventually produces all possible literature by free association. We already have the appropriate tool to do so:“The machinery that allows one to produce an infinite text with a final number of elements has existed for millennia, and this is the alphabet.”[12] All possible combinations of the alphabet would create an infinite library (writing machine of Lagado!). Not only Jonathan Swift imagined such a library. Jorge Luis Borges, an Argentine writer and librarian, wrote a short story about a huge library with an indefinitely large collection of books of uniform format. His detailed description is based on a analytic examination of this fictional library with its enormous scale. The order and content of the books is random and without any meaning. The books contain every possible ordering of the alphabet that is why the majority of the books are pure gibberish. But the library also contains every coherent book that is written or might ever will be. Borges reasons that since all books exist in the library, somewhere one of the books, must be a perfect index of all books of the library. That would be a messianic book, the key to universal knowledge.

---

[11] (Translated from German), Stachowiak, Herbert. p. 87.
[12] Simanowski, Roberto.“Digital Literature”. p. 33.

An objective thought

Since the construction of the early computers, there is a anthropomorphic tendency for names that can be associated with human activities or parts of the human body. It seems as if the computer should imitate everything the human does. Running, reading/writing-head, memory, performance, master/slave, server/ client... etc.. This vocabulary seems to describe a human like system. But with the ideal attribute of pure objectivity, a position a human is very unlikely to reach. That is why the nature of the system is based on algorithms, which are "sequences of formal rules operating on symbolic representations (codes)"[13], to ensure precision and objectivity. On the other hand it tries to imitate the neuronal structures of the human brain. One of the main goals of AI and computer design, is the imitation of this structure. As Raymond Kurzweil, computer scientist and author, proclaims: "By 2030, we'll have the means to scan the human brain and re-create its design electronically."[14] Such a complex structure could have the ability to generate a thought on its own. It could create eternal intellectual work that can be reproduced

---

13 radicalart.info/AlgorithmicArt/intro.html

14 Kurzweil, Raymond. "Live Forever – Uploading The Human Brain... Closer Than You Think",
Available from: www.psychologytoday.com/features3.html

forever, because it is implemented in the system.

3

Automatic text generators

"To give the essence of proposition means to give the essence of all description, therefore the essence of the world. [...] The description of the most general propositional form is the description of the one and only general primitive sign in logic."[15]

Text generators are information processing programs, whose generic processes are defined by stochastic and logic. With the help of mathematical means and statistics, the method of dissection of text is reversed to the technical method of assembling. That is why the structure of automated text corresponds to the design of the program. Algorithmic literature is programmed language. The generated text should achieve the minimum

---

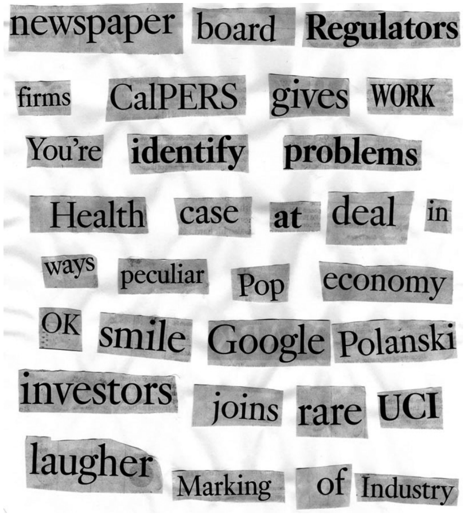
[15] Wittgenstein, Ludwig. p.72

requirement of grammar and semantic comprehensibility. As mentioned in the introduction, the meaning (context) of the compound sentences depends on the repertoire of words – the input material, that is stored in the computer.

3.1
How to make a Dadaist poem

1. Take a newspaper.
2. Take a pair of scissors.
3. Choose an article as long as you are planning to make your poem.
4. Cut out the article.
5. Then cut out each of the words that make up this article and put them in a bag.
6. Shake it gently.
7. Then take out the scraps one after the other in the order in which they left the bag.
8. Copy conscientiously.
9. The poem will be like you.
   And here you are a writer, infinitely original and endowed with a sensibility that is charming though beyond the understanding of the vulgar.

In 1921, Tristan Tzara gave the instruction to cut the words of a selected newspaper's article, his input material, and to shake the scraps in a bag. By repeated pulls of the scraps out of the bag one could create a poem that is an accidental sequence of of words (FIG. 4).



newspaper board Regulators
firms CalPERS gives WORK
You're identify problems
Health case at deal in
ways peculiar Pop economy
OK smile Google Polanski
investors joins rare UCI
laugher Marking of Industry

(FIG. 4)

That is why this instruction functions as a random text generator. Since the emergence of computers, programmers have tried to simulate chance operations with algorithms. But computer generated "chance operations" are just an imitation because they are linked to numbers. However, the final random text gives no

key to decipher it. It may be aesthetically relevant but the spontaneous associations of the observer have no connexion with the author. The instruction could be an attempt to find a new "poetic" language but Tzara's technique creates all sorts of random noise with no meaning. All elements of the text have the same probability. The difference between information and noise fades, that makes all created texts equal in their meanings. The probability to get an informative text with a logic structure is so tiny that the lengths of a human life would not be enough to try. That is like the attempt to understand a text, produced by a monkey[16], who hits keys at random on a typewriter keyboard (in this example the keys would contain words instead of letters).

---

[16] *Infinite monkey theorem:* "Six eternal apes, randomly striking the keys of six eternal typewriters with unlimited amounts of paper and ink would be able to produce Shakespearean sonnets, complete books, and the 23rd Psalm. In the same way, molecular movement, given enough time and matter, could produce Bishop Wilberforce himself, purely by chance and without the work of any designer or creator." Thomas Henry Huxley, in a debate with Samuel Wilberforce, Anglican Bishop of Oxford, about Darwin's theory of natural evolution. Oxford, June 30, 1860.

With the last sentence of his instruction Tzara seems to state ironically the concerns about creativity and individual style of literary production that can be realized by everyman. In fact with this verbal instruction the reader can become a producer whenever he wants to be one. Is the producer of the text and the author the same? This early text-generator demonstrates how conventional text creation, that needs "inner" production rules, can be replaced by processing existing material through "outward" instructions.

3.2
Stochastic Text

In 1959 Theo Lutz executed a program on the electronic mainframe Zuse Z 22 at the T. H. Stuttgart computer centre. The program generated stochastic sentences where the words are determined randomly. The machine stores a certain number of subjects, predicates, logical operators ("A", "AN", "EVERY", "NO", "NOT EVERY") and logical constants ("AND", "OR", "THEREFORE", "."), coded as binary numbers. Using the first random number the machine forms the position number in the store of a subject. In the next step the program locates a code number which defines the gender of the subject (0 = masculine, 1 = feminine and 2 = neutral). The machine then determines (randomly) a logical operator and coordinates this with the gender of the subject, using the located code number. Theo Lutz selected 16 subjects and 16 predicates from Franz Kafka's "The Castle" as original material. With this input the program can form 4,174,304 different pairs of elementary sentences.

Subjects and predicates from Kafka's "The Castle":

the count,
the stranger,
the look,
the church,
the castle,
the picture,
the eye,
the village,
the tower,
the farmer,
the way,
the guest,
the day,
the house,
the table,
the labourer

open,
silent,
strong,
good,
narrow,
near,
new,
quiet,
far,
deep,
late,
dark,
free,
large,
old,
angry

A selection of stochastic sentences[17]:

Not every look is near.
No village is late.
A castle is free and every farmer is far.
Every stranger is far.
A day is late.
Every house is dark.
An eye is deep.
Not every castle is old.
Every day is old.
Not every guest is angry: a church is narrow.
No house is open and not every church is silent.
Not every eye is angry.
No look is new.
Every way is near.
Not every castle is quiet.
No table is narrow and every tower is new.
Every farmer is free.
Every farmer is near.
No way is good or not every count is open.

---

[17] (Translation by Helen MacCormack, 2005) In: *augenblick* 4 (1959), No. 1, p. 3–9

Compared with Tzaras method, Theo Lutz also“cut out” the words of a given text and arranged them in a new order. But Lutzs system achieves comprehensibility due to basic grammatical structure and predefined material. His stochastic text creates a Kafkaesque atmosphere that is strange to read. But the created sentences are very repetitive and after a few lines one gets tired of reading further. There is no variance because the concept of the program code is too simple and the input material is too limited. So what is the purpose of thousand similar sentences? Nevertheless, this program can be seen as a warm-up for the following text generators. It’s limitations where obvious, so one had to figure out how to create more variety, more alternatives and tension.

3.3
Romance Writer

An example for a more advanced text generator is Nick Sullivans Romance Writer from 1997, which creates automatically generated stories. Sullivan chose his input material from Regency romances, a subgenera of romantic novels from the early 19th century. The program code is based on different variables that represent a general category (FIG. 5). Variables are like place holders. They have a fixed position in a given text and can be randomly filled with content. Like *[signal]* could be filled with“a knock at the door”,“an abrupt fanfare”, or “the sound she had been longing to hear”. It is possible to create a more interesting variety of stories if the chosen material is interesting in itself... with an exaggerate expression. Theo Lutz used word material which was already connected by meaning due to Kafkas text, but the words have no specific attraction in itself. By contrast, Sullivans text depends in large parts on the meaning of its parts“[...] and if it has no sense this can only be because we have given no meaning to some of its constituent parts."[18]

Program structure:

[title]

Waiting alone in the **[wait place]**, with the **[waft-item]** wafting in from outside, **[heroine]** thought once more of **[hero]**, the **[hero-description]**. He was now, according to **[informant]**, **[hero activity]**.

Then came **[signal]**, and she **[reaction]**. He was here! "**[fond-remark]**, **[pet-name]**" he **[said]**, **[embrace]**, and as **[finale]**, she **[denouement]**.

---

[18] Wittgenstein, Ludwig, p.73

Romance Writer
example #1

Incense and Nobility

Waiting alone in the little walled garden, with the far-off clamour of the playing fields wafting in from outside, Serena thought once more of former Prime Minister Pitt the Younger, the mysterious stranger with the large dog. He was now, according to father, studying medicine in Zurich with Dr. Frankenstein.

Then came the sound she had been longing to hear, and she whirled around. He was here! "I've thought of you every minute I've been away, you little fool!" he breathed, drawing her to him as a moth to a lantern wick, and as the musicians began to play, she wordlessly let her body melt against his own.

example #2

Recompense and Prejudice

Waiting alone in the gleaming ballroom, with the aroma of fresh mulberry pies wafting in from outside, Elizabeth, young governess, thought once more of Reginald Fortescue, the loving suitor she had turned so thoughtlessly away. He was now, according to the best salon gossip, a prisoner in the very castle he had once owned.

Then came the thud of fists, a muffled cry, fighting on the stairway, and she dropped the brimming wine glass heedlessly on the rug. He was here! “Even Lady Louisa couldn’t keep me from you, my giddy little goose!” he said with his eyes (for his voice was now silent for ever), passion igniting his strong features, and as he once again began to woo her with the sensual voice of his Stradivarius, she began to wonder how she would word the wedding announcement.

example #3

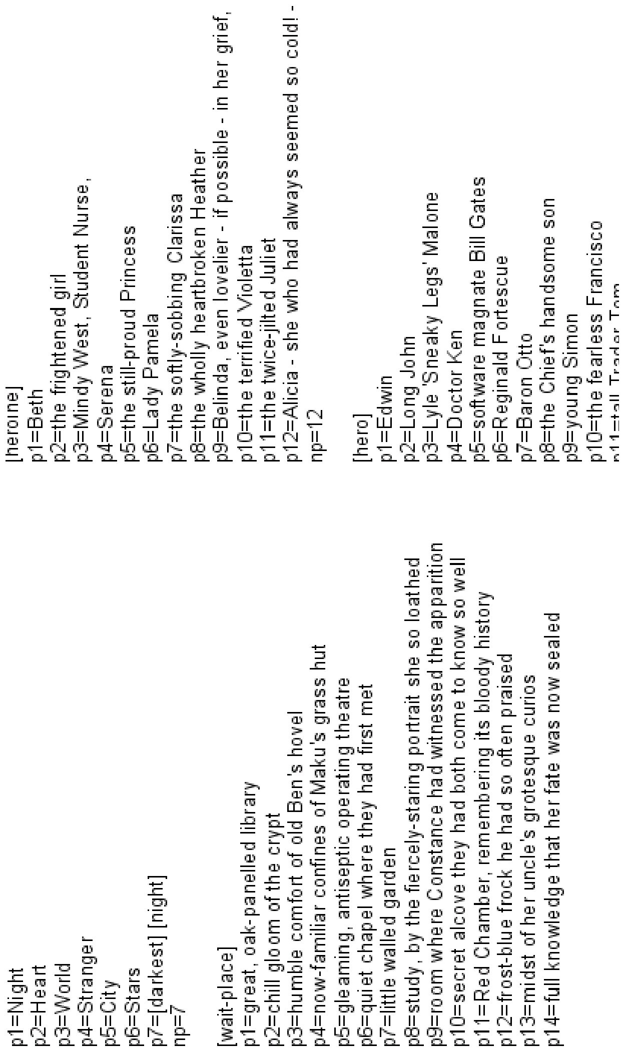
Noble Heiress

Waiting alone in the midst of her uncle’s grotesque curios, with the fragrance of a new spring wafting in from outside, the frightened heiress thought once more of Lord Byron, the bold aristocrat who had been so gentle in her arms. He was now, according to Sir Walter Elliot, on the mail coach to London, leaving her world forever.

Then came a knock at the door, and she rose to face the inevitable. He was here! “Come to me, my perfect little carrot!” he stammered, his eyes alive in the slanting light, and as the music in her heart rose to a new crescendo of happiness, she began to wonder how she would word the wedding announcement.

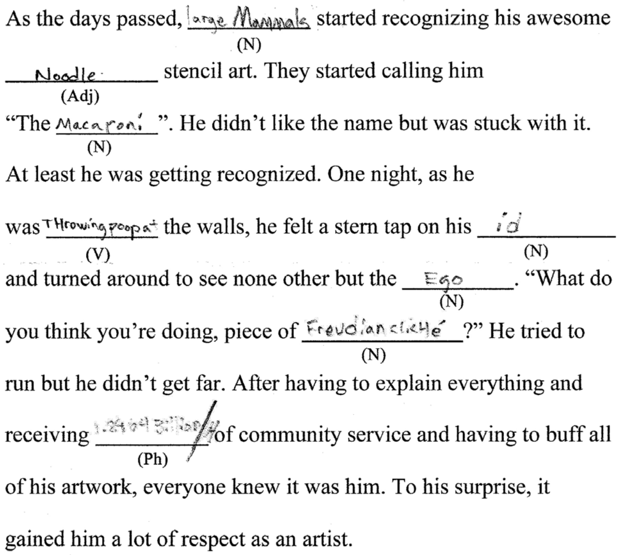
If the given input material is too neutral, the simple structure of the program would become too obvious. The quality of the final text depends also on fortunate coincidence – some doesn’t make sense but others are unique.

Sullivans text generator features the same structure than Mad Libs, a phrasal template word game, that was invented in 1953 by Leonard Stern and Roger Price (FIG. 6). This game consist of short stories with many key words replaced with blanks. Beneath each blank is specified a lexical or other category, such as“noun”,“verb”,“place”, or“part of the body”. One player asks the other players, in turn, to contribute some word for the specified type for each blank, but without revealing the context for that word.



```
p1=Night
p2=Heart
p3=World
p4=Stranger
p5=City
p6=Stars
p7=[darkest] [night]
np=7

[wait-place]
p1=great, oak-panelled library
p2=chill gloom of the crypt
p3=humble comfort of old Ben's hovel
p4=now-familiar confines of Maku's grass hut
p5=gleaming, antiseptic operating theatre
p6=quiet chapel where they had first met
p7=little walled garden
p8=study, by the fiercely-staring portrait she so loathed
p9=room where Constance had witnessed the apparition
p10=secret alcove they had both come to know so well
p11=Red Chamber, remembering its bloody history
p12=frost-blue frock he had so often praised
p13=midst of her uncle's grotesque curios
p14=full knowledge that her fate was now sealed

[heroine]
p1=Beth
p2=the frightened girl
p3=Mindy West, Student Nurse,
p4=Serena
p5=the still-proud Princess
p6=Lady Pamela
p7=the softly-sobbing Clarissa
p8=the wholly heartbroken Heather
p9=Belinda, even lovelier - if possible - in her grief,
p10=the terrified Violetta
p11=the twice-jilted Juliet
p12=Alicia - she who had always seemed so cold! -
np=12

[hero]
p1=Edwin
p2=Long John
p3=Lyle 'Sneaky Legs' Malone
p4=Doctor Ken
p5=software magnate Bill Gates
p6=Reginald Fortescue
p7=Baron Otto
p8=the Chief's handsome son
p9=young Simon
p10=the fearless Francisco
p11=tall Trader Tom
```

FIG. 5 Romance Writer, code excerpt



As the days passed, large Mammals (N) started recognizing his awesome
Noodle (Adj) stencil art. They started calling him
“The Macaroni (N)”. He didn’t like the name but was stuck with it.
At least he was getting recognized. One night, as he
was THrowing poop at (V) the walls, he felt a stern tap on his id (N)
and turned around to see none other but the Ego (N). “What do
you think you’re doing, piece of Freudian clicHé (N)?” He tried to
run but he didn’t get far. After having to explain everything and
receiving 1.24649 Billion/H (Ph) of community service and having to buff all
of his artwork, everyone knew it was him. To his surprise, it
gained him a lot of respect as an artist.

FIG. 6 Mad Libs

3.4
RACTER

> “With the exception of this introduction, the writing in this book was all done by a computer. [...] Computers are supposed to compute. They are designed to accomplish in seconds (or microseconds) what humans would require years or centuries of concerted calculation effort to achieve. They are tools we employ to get certain jobs done. [...] Prose is the formal communication of the writer’s experience, real and fancied. But, crazy as this may sound, suppose we remove that criterion: suppose we somehow arrange for the production of prose that is in no way contingent upon human experience. What would that be like? Indeed, can we even conceive of such a thing?”

William Chamberlain, RACTER,
The Policeman’s Beard is Half Constructed

RACTER, which is short for “raconteur”, is a program written by William Chamberlain and Thomas Etter in 1983. The program was executed on the Zuse Z80 micro with 64K of RAM. It generated grammatically correct prose through“ free association” (text templates). As a “proof” of their concept, they published a book using nothing but the output from the program. That book was “The Policeman’s Beard Is Half Constructed”. (FIG. 7–10) The writing style of this book is very mysterious and surreal and doesn’t follow any conventions. Joan Hall illustrated the text, using a combination of computer graph art and antique portraits. Her illustrations correlate quiet well with the strange, poetic sentences.

> **RACTER:**
>
> **More than iron, more than lead, more**
> **than gold I need electricity.**
> **I need it more than I need lamb or pork**
> **or lettuce or cucumber.**
> **I need it for my dreams.**

Even so RACTER produces a range of interesting texts that come quite close to

surrealist poems[19], on some pages the algorithm still gleams through the text and unmasks the program. There is for example a collection of limericks (FIG. 7) that seem to display the limitations of RACTER quite strongly. The first line almost always begins with an adjective-adjective-object and certain rhymes and words are repeated constantly.“It is pretty clear that Chamberlain’s language constructor code is crude, deliberate, and limited, that it extensively leans upon human pre-written templates, random word selection, and object/subject tracking. The fact that we, RACTER’s audience, are so willing to prop up and fill in any and all missing context, coherence, and relevance is interesting in itself. It all makes me more certain that true advances in AI

---

[19] For example Paul Eluards “Max Ernst” (*Captial of Pain*, 1926)

[...]
In the brightest corner of every eye
We’re expecting the fish of anguish.
In a corner the car of summer
Immobile glorious and forever.
[...]

will come [...] when we humans remove ourselves completely from the fitness metric"[20].

RACTER's saga doesn't end with the book. A commercial version for the home computer was published in 1984. This version would take words from the user input and tried to use them as frequently as possible in conversation, which did not allow for much variety. The commercial version failed and some people even accused Chamberlain of heavily modifying the original RACTER output to make it sound more plausible. A much more powerful interactive program, that was written by Joseph Weizenbaum in 1966, was ELIZA (see 3.5, page 57).

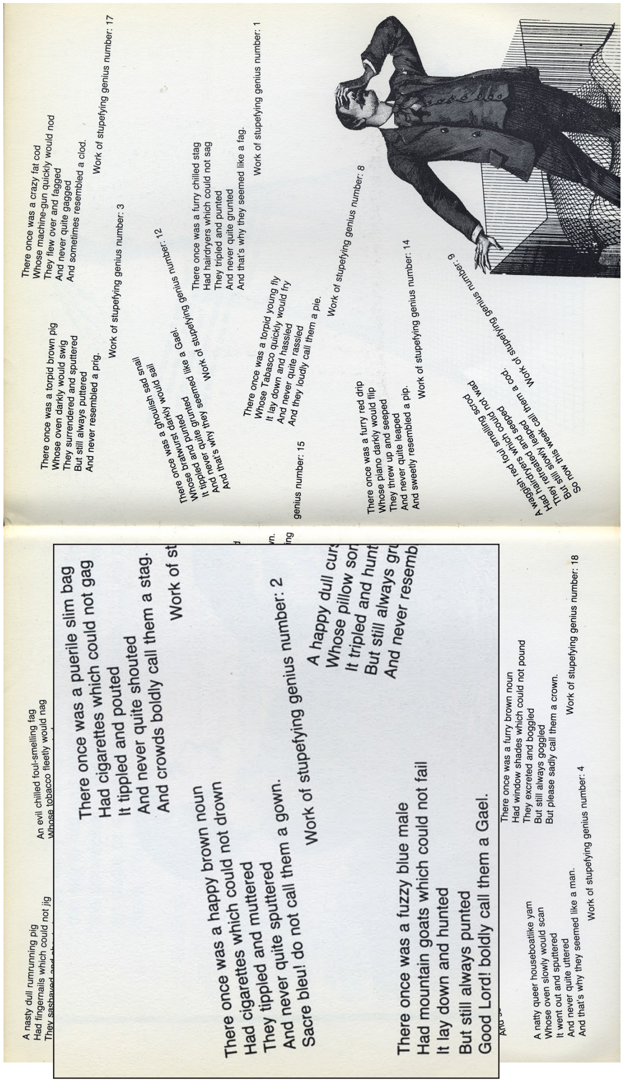


FIG. 7 RACTER, selection of limericks

A nasty dull rumrunning pig
Had fingernails which could not jig

An evil chilled foul-smelling fag
Whose tobacco fleetly would nag

There once was a puerile slim bag
Had cigarettes which could not gag
It tippled and pouted
And never quite shouted
And crowds boldly call them a stag.

Work of st

There once was a happy brown noun
Had cigarettes which could not drown
They tippled and muttered
And never quite sputtered
Sacre bleu! do not call them a gown.

Work of stupefying genius number: 2

A happy dull curs
Whose pillow son
It tripled and hunt
But still always gru
And never resemb

There once was a fuzzy blue male
Had mountain goats which could not fail
It lay down and hunted
But still always punted
Good Lord! boldly call them a Gael.

A natty queer houseboatlike yam
Whose oven slowly would scan
It went out and sputtered
And never quite uttered
And that's why they seemed like a man.

Work of stupefying genius number: 4

There once was a furry brown noun
Had window shades which could not pound
They excreted and boggled
But still always goggled
But please sadly call them a crown.

Work of stupefying genius number: 18

There once was a torpid brown pig
Whose oven darkly would swig
They surrendered and sputtered
But still always puttered
And never resembled a prig.

Work of stupefying genius number: 3

There once was a crazy fat cod
Whose machine-gun quickly would nod
They flew over and fagged
And never quite gagged
And sometimes resembled a clod.

Work of stupefying genius number: 17

There once was a ghoulish sad snail
Whose bratwurst darkly would sail
It tippled and punted
And never quite grunted
And that's why they seemed like a Gael.

Work of stupefying genius number: 12

There once was a furry chilled stag
Had hairdryers which could not sag
They tripled and punted
And never quite grunted
And that's why they seemed like a fag.

Work of stupefying genius number: 1

genius number: 15

There once was a torpid young fly
Whose Tabasco quickly would fry
It lay down and hassled
And never quite rassled
And they loudly call them a pie.

Work of stupefying genius number: 8

There once was a furry red drip
Whose piano darkly would flip
They threw up and seeped
And never quite leaped
And sweetly resembled a pip.

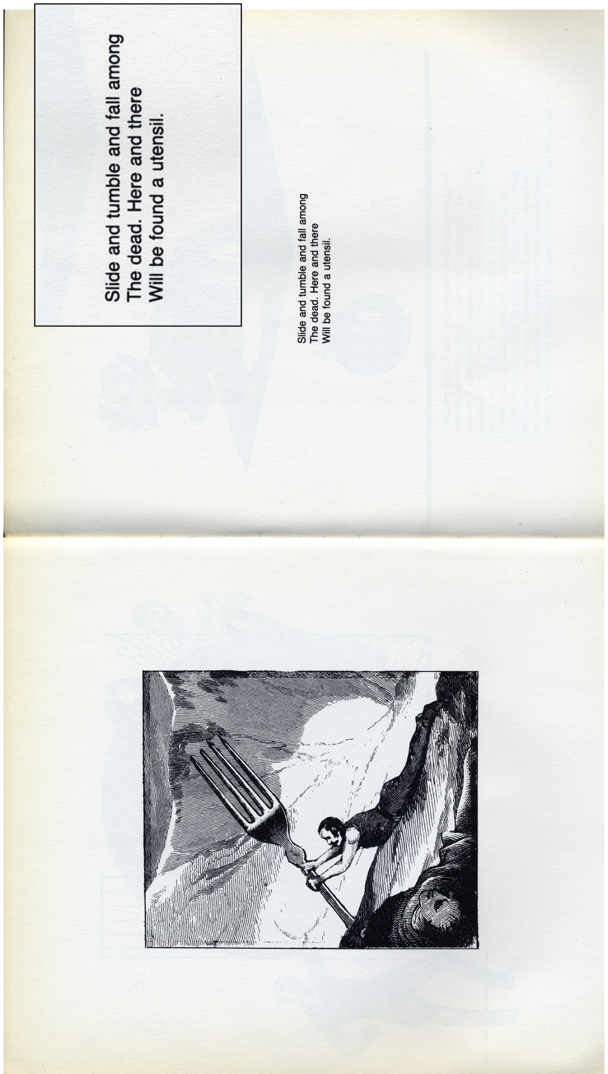
Work of stupefying genius number: 14

A waggish red foul smelling scrod
Had hairdryers which could not wad
They retreated and seeped
But still slowly leaped
So now this week call them a cod.

Work of stupefying genius number: 9

---

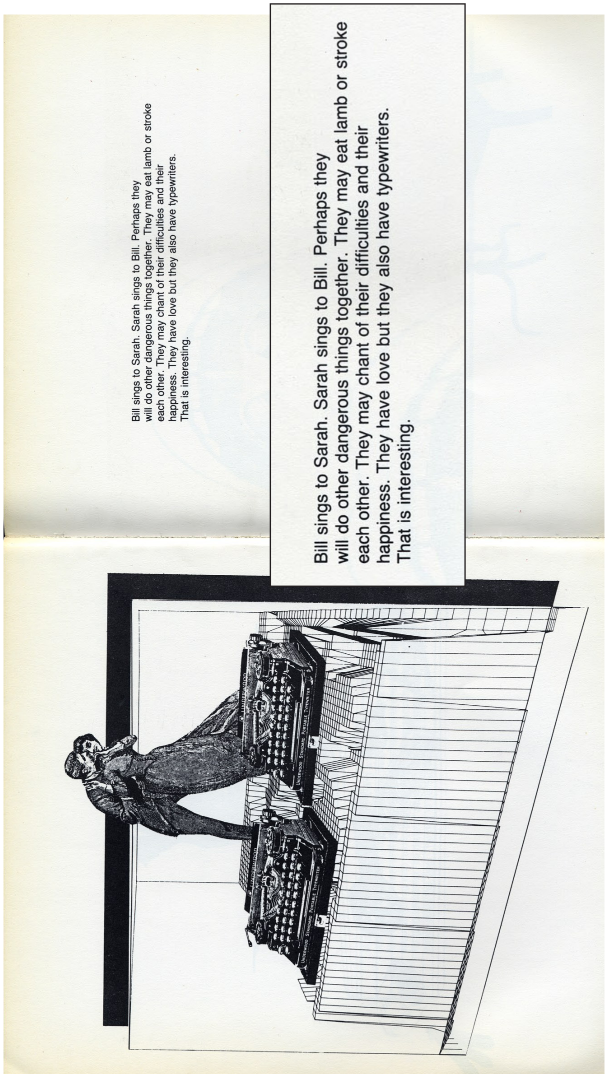
[20] Reetz, Randall. *Bay Area Artificial Intelligence Meet up Group,* thread, Available from: www.ai-meetup.org, 2010. Randell possibly points out that some advanced computer programs are able to generate their own algorithms (genetic algorithms) and select the best results on their own (survival of the fittest). This is called evolutionary computing.



Slide and tumble and fall among
The dead. Here and there
Will be found a utensil.

Slide and tumble and fall among
The dead. Here and there
Will be found a utensil.

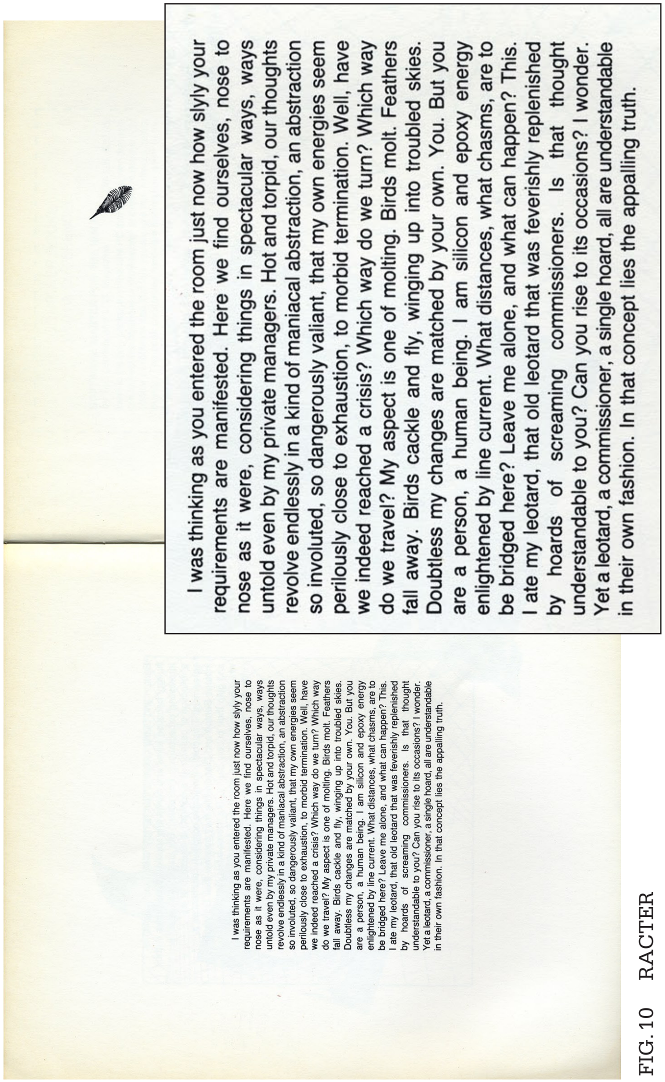
FIG. 8 RACTER



Bill sings to Sarah. Sarah sings to Bill. Perhaps they
will do other dangerous things together. They may eat lamb or stroke
each other. They may chant of their difficulties and their
happiness. They have love but they also have typewriters.
That is interesting.

Bill sings to Sarah. Sarah sings to Bill. Perhaps they
will do other dangerous things together. They may eat lamb or stroke
each other. They may chant of their difficulties and their
happiness. They have love but they also have typewriters.
That is interesting.

FIG. 9 RACTER



I was thinking as you entered the room just now how slyly your requirements are manifested. Here we find ourselves, nose to nose as it were, considering things in spectacular ways, ways untold even by my private managers. Hot and torpid, our thoughts revolve endlessly in a kind of maniacal abstraction, an abstraction so involuted, so dangerously valiant, that my own energies seem perilously close to exhaustion, to morbid termination. Well, have we indeed reached a crisis? Which way do we turn? Which way do we travel? My aspect is one of molting. Birds molt. Feathers fall away. Birds cackle and fly, winging up into troubled skies. Doubtless my changes are matched by your own. You. But you are a person, a human being. I am silicon and epoxy energy enlightened by line current. What distances, what chasms, are to be bridged here? Leave me alone, and what can happen? This. I ate my leotard, that old leotard that was feverishly replenished by hoards of screaming commissioners. Is that thought understandable to you? Can you rise to its occasions? I wonder. Yet a leotard, a commissioner, a single hoard, all are understandable in their own fashion. In that concept lies the appalling truth.

FIG. 10 RACTER

3.5

ELIZA: “Does it please you to believe I am afraid of you?”

Joseph Weizenbaum, a German computer scientist, named his dialogue program ELIZA, as a reference to Bernhard Shaws *Pygmalion.*[21] Weizenbaum researched at the MIT in the field of natural language processing. His program ELIZA simulates a psychologist. Weizenbaum modelled its conversational style after Carl Rogers, a famous psychologist who introduced the method of open-ended questions to encourage patients to communicate more effectively with therapists. Weizenbaum chose the name ELIZA to draw attention to the seemingly infinite perfectibility of the software: “These programs, like the Eliza of the Pygmalion fame, can be taught to speak increasingly well.”[22]

The program reacts on each user input with a single sentence by recognition

---

[21] Also called *My Fair Lady*, Musical from 1964. Audrey Hepburn played “Eliza” and became world famous.

[22] Link, David. p. 35

of key words or phrases. With the help of a key word list, the algorithm examines the input of the user. The words are sorted by importance, that means the program reacts on some words first (FIG. 11). Responses are generated by reassembly rules associated with selected decomposition rules. Thus an illusion of understanding is generated. ELIZA showed that such an illusion is surprisingly easy. For example if the user talks about his mother, ELIZA will ask about the family. These conversational responses are interpreted as intelligent. They are at the same time very vague and nonspecific. That is why ELIZA performs best when the user "talks" just as he would to a psychiatrist.

The fundamental technical problems with which ELIZA is concerned are[23]:

- The identification of key words.
- The discovery of minimal context.
- The choice of transformations.
- Generation of responses in the absence of keywords, and
- The provision of an ending capacity for ELIZA "scripts".

A typical conversation is the following:

User: Men are all alike.

ELIZA: IN WHAT WAY?

User: They're always bugging us about something or other.

ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE?

User: Well, my boyfriend made me come here.

ELIZA: YOUR BOYFRIEND MADE YOU COME HERE?

User: He says I'm depressed much of the time.

ELIZA: I AM SORRY TO HEAR YOU ARE DEPRESSED.

---

[23] Weizenbaum, Joseph

User: It’s true. I am unhappy

ELIZA: DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY?

User: I need some help, that much seems certain.

ELIZA: WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP?

User: Perhaps I could learn to get along with my mother.

ELIZA: TELL ME MORE ABOUT YOUR FAMILY.

User: My mother takes care of me.

ELIZA: WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?

User: My father.

ELIZA: YOUR FATHER.

User: You are like my father in some ways.

ELIZA: WHAT RESEMBLANCE DO YOU SEE?

User: You are not very aggressive but I think you don’t want me to notice that.

ELIZA: WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE.

User: You don’t argue with me.

ELIZA: WHY DO YOU THINK I DON’T ARGUE WITH YOU?

User: You are afraid of me.

ELIZA: DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU?

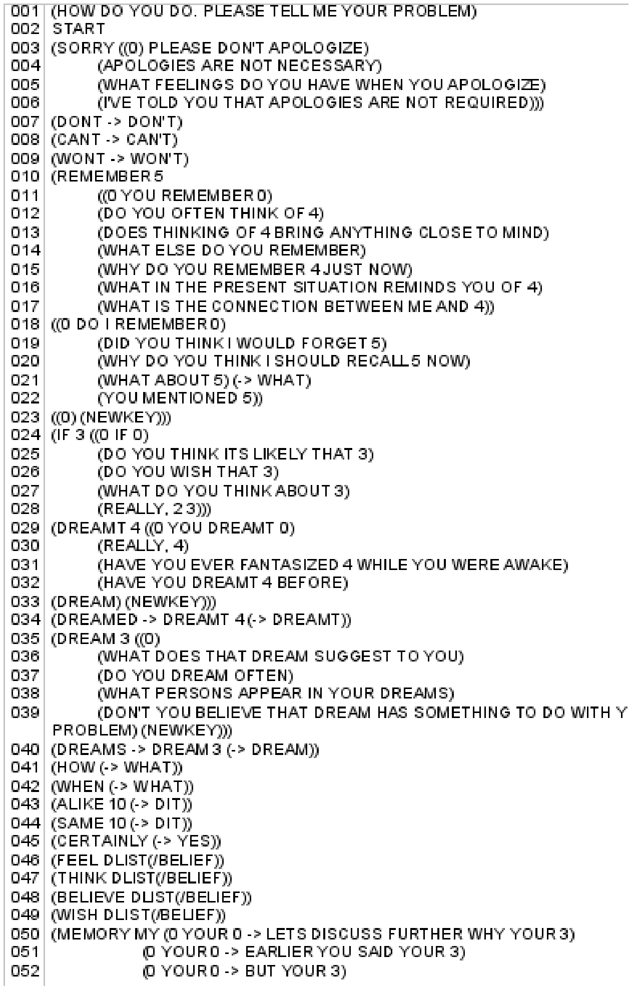
User: My father is afraid of everybody.

ELIZA: WHAT ELSE COMES TO YOUR MIND WHEN YOU THINK OF YOUR FATHER?

User: Bullies.

ELIZA: DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE?

ELIZA was the first programmed chatbot[24] ever (1966), followed by PERRY (Kenneth Colby, 1972), a simulation of a paranoid schizophrenic, RACTER (only the interactive version), Jabberwock (Jürgen Pirner, 1997), a humorous chatbot, inspired by the nonsense verse poem“Jabberwocky” written by Lewis Caroll, and A.L.I.C.E (Richard Wallace, 1998) an awarded program, that is considered to be the most human-like chatbot due to the definitions of the Turing test.[25] As much as these chatbots operate at its best within a predefined subject, automatic text generators simply produce the best results if there is a theme. Like betrayal, the special subject of BRUTUS.



```
(HOW DO YOU DO. PLEASE TELL ME YOUR PROBLEM)
START
(SORRY ((0) PLEASE DON'T APOLOGIZE)
        (APOLOGIES ARE NOT NECESSARY)
        (WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE)
        (I'VE TOLD YOU THAT APOLOGIES ARE NOT REQUIRED)))
(DONT -> DON'T)
(CANT -> CAN'T)
(WONT -> WON'T)
(REMEMBER 5
        ((0 YOU REMEMBER 0)
        (DO YOU OFTEN THINK OF 4)
        (DOES THINKING OF 4 BRING ANYTHING CLOSE TO MIND)
        (WHAT ELSE DO YOU REMEMBER)
        (WHY DO YOU REMEMBER 4 JUST NOW)
        (WHAT IN THE PRESENT SITUATION REMINDS YOU OF 4)
        (WHAT IS THE CONNECTION BETWEEN ME AND 4))
((0 DO I REMEMBER 0)
        (DID YOU THINK I WOULD FORGET 5)
        (WHY DO YOU THINK I SHOULD RECALL 5 NOW)
        (WHAT ABOUT 5) (-> WHAT)
        (YOU MENTIONED 5))
((0) (NEWKEY)))
(IF 3 ((0 IF 0)
        (DO YOU THINK ITS LIKELY THAT 3)
        (DO YOU WISH THAT 3)
        (WHAT DO YOU THINK ABOUT 3)
        (REALLY, 2 3)))
(DREAMT 4 ((0 YOU DREAMT 0)
        (REALLY, 4)
        (HAVE YOU EVER FANTASIZED 4 WHILE YOU WERE AWAKE)
        (HAVE YOU DREAMT 4 BEFORE)
(DREAM) (NEWKEY)))
(DREAMED -> DREAMT 4 (-> DREAMT))
(DREAM 3 ((0)
        (WHAT DOES THAT DREAM SUGGEST TO YOU)
        (DO YOU DREAM OFTEN)
        (WHAT PERSONS APPEAR IN YOUR DREAMS)
        (DON'T YOU BELIEVE THAT DREAM HAS SOMETHING TO DO WITH Y
PROBLEM) (NEWKEY)))
(DREAMS -> DREAM 3 (-> DREAM))
(HOW (-> WHAT))
(WHEN (-> WHAT))
(ALIKE 10 (-> DIT))
(SAME 10 (-> DIT))
(CERTAINLY (-> YES))
(FEEL DLIST(/BELIEF))
(THINK DLIST(/BELIEF))
(BELIEVE DLIST(/BELIEF))
(WISH DLIST(/BELIEF))
(MEMORY MY (0 YOUR 0 -> LETS DISCUSS FURTHER WHY YOUR 3)
                (0 YOUR 0 -> EARLIER YOU SAID YOUR 3)
                (0 YOUR 0 -> BUT YOUR 3)
```

FIG. 11 A sampling from an ELIZA script

---

[24] from *chatter* and ro-*bot*

[25] Turing test: The test was introduced by Alan Turing in his 1950. It tests a machine’s intelligence. A human judge engages in a conversation, limited to a text-only channel, with a human and a machine. All participants are separated from one another. If the judge cannot reliably tell the machine from the human, the machine is said to have passed the test. Since today, no program has passed this test.

3.6
BRUTUS

BRUTUS is the most recent program of the introduced text generators. It is named after the betrayer Marcus J. Brutus, who took a leading role in the assassination of Julius Caesar. It's inventors, Selmer Bringsjord and David A. Ferrucci, finished it in 2000, after seven years of development work. The final program writes short stories with around 500 words about predefined characters, goals, events and places. The system is based on a simple rule: a figure X betrays a figure Y but only if there exist a positive value Z in which figure Y is enormously interested in.

BRUTUS, example of a short story:

Dave Striver loved the university. He loved its ivy-covered clock towers, its ancient and sturdy brick, and its sun-splashed verdant greens and eager youth. He also loved the fact that the university is free of the stark unforgiving trials of

the business world – only this isn't a fact: academia has its own tests, and some are as merciless as any in the marketplace. A prime example is the dissertation defence: to earn the PhD, to become a doctor, one must pass an oral examination on one's dissertation. This was a test Professor Edward Hart enjoyed giving. Dave wanted desperately to be a doctor. But he needed the signatures of three people on the first page of his dissertation, the priceless inscriptions which, together, would certify that he had passed his defence. One of the signatures had to come from Professor Hart, and Hart had often said – to others and to himself – that he was honoured to help Dave secure his well-earned dream. Well before the defence, Dave gave Hart a penultimate copy of his thesis.

The professor read it and told Dave that it was absolutely first-rate, and that he would gladly sign it at the defence. They even shook hands in Hart's book-lined office. Dave noticed that Hart's eyes were bright and trustful, and his bearing paternal.

At the defence, Dave thought that he eloquently summarized Chapter 3 of his dissertation. There were two questions, one from Professor Rogers and one from Dr. Teer; Dave answered both,

apparently to everyone's satisfaction. There were no further objections. Professor Rogers signed. He slid the tome to Teer; she too signed, and then slid it in front of Hart. Hart didn't move."Ed?" Rogers said. Hart still sat motionless. Dave felt slightly dizzy. "Ed, are you going to sign?" Later, Hart sat alone in his office, in his big leather chair, saddened by Dave's failure. He tried to think of ways he could help Dave achieve his dream.

The fundamental questions, with which the inventors where concerned, are:

- How the system progress and maintains coherence of the whole story?
- How does it ensure novelty and interestingness?
- How does it create suspense?
- How to generates natural language structures and meaning?

The story reads well, the language sounds natural and nothing seems to be randomly combined, everything makes sense (apart from the last sentence). Minimum requirements on a readable story.

But something is missing. This story is not satisfying! It should generate suspense and tension. It seems that the topic "betrayal" is not sufficient enough to write a good story. Even if Bringsjord spend two years, to analyse how betrayal can be programmed – BRUTUS remains a limited writer. The only setting it describes is academia. Bringsjord and Ferrucci had the ambitious plan to figure out what makes a story interesting from the standpoint of logic, then to formalize interestingness in some logical system, and finally code this formalization. "This idea proved to be a painful dead end."[26] After seven years of research, these scientists finally agreed, that it was not possible to program interesting stories.

---

[26] Bringsjor, Selmer (1999). *Brutus and the Narrational Case Against Church's Thesis*. From: AAAI Technical Report FS-99-01

4
Conclusion

Simulated creativity

With the enhancement and the perfecting of text generators rises the question if algorithmic literature could have the same artistic value in literature than for example James Joyce's *Ulysses*. Or even more drastic: Can an individual author be imitated and finally replaced by a machine? Even if it would be possible to recreate the design of the brain electronically, this would only imitate the mind in general. That means only the parts which are identical within every brain. This formality does not consider the individuum.

But can a machine become creative in writing? Or rather, can they at least appear to be creative? Margaret Boden, researcher in the fields of AI and Cognitive Science, defines creativity as "the ability to come up with ideas or artefacts that are new, surprising, and valuable."[27] But what is a new idea? It could be new to the person who has it (even so the idea already exists) or no one else has had it before and it has rise for the first time in human history. We hardly can imagine that a computer program would be capable to come up with a groundbreaking new idea on its own. The computer may reproduce a lot of thoughts but this depends on the author who implements the algorithm. To have a pioneering idea it misses the right input data because thinking is also based on experiences and the frame of reference – the computer can not create this on his own.

Boden describes three main ways of creativity. The first involves making unfamiliar combinations of familiar ideas. The second involves the exploration, and the third the transformation, of"conceptual spaces"[28] (structured styles of thought) in people's minds.

---

[27] Boden, Margaret A. pp. 1–10

[28] ibid.

Combine

For a computer it is easy to pick out two ideas (two data-structures) and combine them. But would the result be of any interest? Combining ideas creatively isn't like shaking Tzaras scraps in a bag. The scraps have to come together because there is some conceptional link between them which we value because it is thought-provoking or humorous (this depends of course on our background knowledge). If Stephen Hawking compares the universe with a nutshell he requires a rich store of knowledge to make this novel combination. But could we imagine a super computer, that stores all the knowledge of the world (Library of Babel!) – a performance a human never would be capable of? Would it not be possible for this computer to make connections between everything, the key to the question "what holds the world together at the core"?[29] It would be very "alone" with this discovery because it is far

---

[29] "Was die Welt im Innersten zusammenhält" Goethe, Johann Wolfgang, *Faust. Der Tragödie Erster Teil*.

too complex for the human brain to understand (which leads to the question: What is the purpose of such a system?).

Explore

What about exploratory creativity? Within a given discipline, for example writing limericks, many ideas are possible, but only some of which may have been actually thought. The space of all possible limericks has not– and never will be exhausted because the number of rhymes is incredible large. If a program like RACTER writes a long list with limericks, it enables the author to explore all the possibilities he hadn't thought about.

Transform

The latest researches in algorithmic literature apply AI technics to simulate the structures and processes of our brain. To transform a thought means to have an apparently impossible idea because the pre-existing way of thinking was changed. AI programs can make random changes

(self transformation) in their current rules so that new forms of structure result. Like a simulated evolution these programs can creates different generations, the "best" structures are selected, and used to breed the next generation. In each case, the selection of the "fittest" is done by a human, so that the program can learn what is good and what is rubbish. The machine isn't conscious, and has no desires or values, so it can not judge what it's doing. But it can remember and learn! And therefore it can imitate.

Remember the case of Adolf Knipe, who never told anybody that all the stories, written by his machine, are just computed plot scenarios. We can easily imagine to get caught in the trap as well. How can we know if a story is written by a human or a machine? Because of its creative value?

Furthermore Margaret Boden stated that ideas are only creative if they are valuable. But what is valuable? Maybe it is even more difficult to define. Because creativity values differ so much from each other, it finally seems that many discussions about creativity originate from disagreements about value. This applies to computer literature no less than to human writing. That is maybe why BRUTUS remained a limited storyteller. Because we don't have objective measurements to explain what is interesting, valuable and creative. But the computer has the appropriate tools to perform this task: "Since neural nets[30] are good at learning to recognize patterns even when we do not really understand how they do it, the argument seems insufficient to establish that machines could not learn to recognize creativity and (in the context of a selectionist system) thereby learn to be creative."[31]

But finally one can say, even if our aesthetic sensation could be identified and coded to enable the computer to develop its own writing styles, there would still be disagreement about whether the computer even appears to be a creative writer.

---

[30] A neural net is an artificial networks, which is composed of artificial neurons (mathematical functions) or nodes. They imitate the structure of our biological neural network (nervous system).

[31] De Sousa, Ronald

Timeline

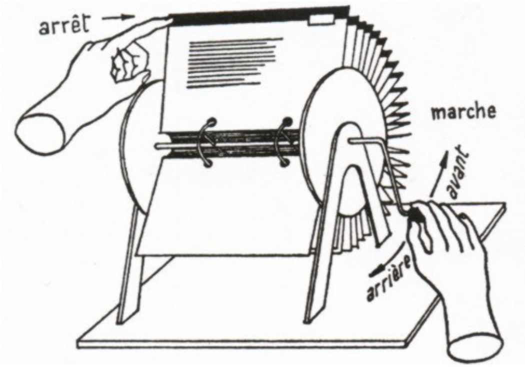
This rough timeline lists some further examples of algorithmic literature, reading machines and text generators from 1920–2002.

1920

- Tristan Tzara – *How to make a Dadaist poem.* Tzara gave the instruction to cut out words from a newspaper to shake them in a bag and then put together a new text.

1937

- Juan-Esteban Fassio constructed a reading machine to assist in reading Raymond Roussel's revision of the codex *Nouvelles Impressions d'Afrique (1932)*to make the complexity of Roussel's writings more comprehensible (his texts consists of endless lists, side notes, footnotes, and parentheses with 9 levels of interlocking reference). The *reading machine* was shown in 1937 at a surrealist exhibition for which he had the text printed on cards. The cards are installed around the axis of a drum (round register), which the reader rotates via a crank. The related text card could be found through an upwardly protruding coloured marker. "These devices [...] activate looping, multidimensional circuits of textual intercomparison, and project the printed page into complex field geometries."

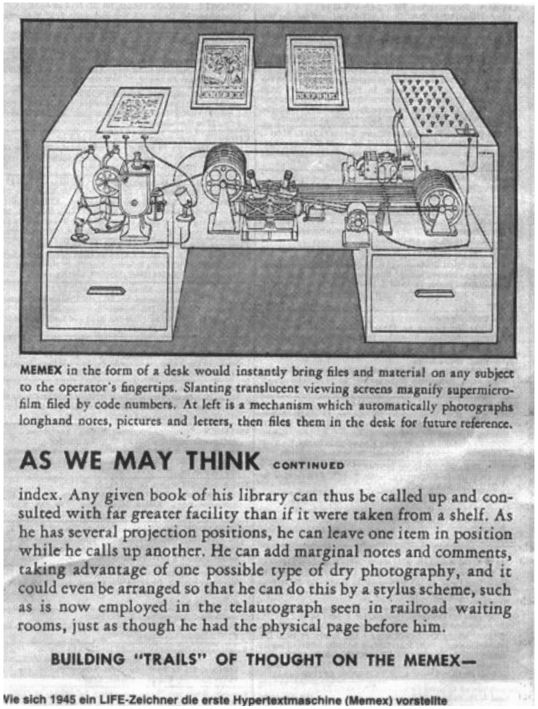


Reading Machine

Harpold, Terry (2009). *Ex-Foliations: Reading Machines and the Upgrade Path.* Minneapolis: University of Minnesota Press, p. 13
Picture available from: netlern.net/hyperdis/txt/alte/htutopia/roussel.htm

1945

- Vannevar Bush published the article *As We May Think*. Explaining the idea of Memex (Memory Extender), a proto-hypertext system which supports associative thinking. Bush envisioned the Memex as a device in which individuals would compress and store all of their books, records, and communications, "mechanized so that it may be consulted with exceeding speed and flexibility".



MEMEX in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference.

AS WE MAY THINK CONTINUED

index. Any given book of his library can thus be called up and consulted with far greater facility than if it were taken from a shelf. As he has several projection positions, he can leave one item in position while he calls up another. He can add marginal notes and comments, taking advantage of one possible type of dry photography, and it could even be arranged so that he can do this by a stylus scheme, such as is now employed in the telautograph seen in railroad waiting rooms, just as though he had the physical page before him.

BUILDING "TRAILS" OF THOUGHT ON THE MEMEX—

Vie sich 1945 ein LIFE-Zeichner die erste Hypertextmaschine (Memex) vorstellte

How the Memex could look like

Bush, Vannevar (July 1945). "As We May Think". *The Atlantic Monthly.*
Article available from: www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/3881/
Picture available from: www.memexplex.org/

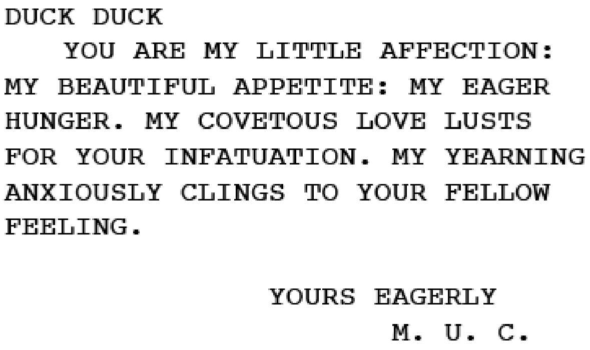
1948

- Claude Shannon generates chance literature with Markov chains. In *A Mathematical Theory of Communication,* Shannon proposed using a Markov chain to create a statistical model of the sequences of letters in a piece of English text.

Text available from: cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

1953

- Christopher Strachey, a physicist, produced love letters with a computer program. The algorithm producing the text was based on a template that randomly selected words from sets that were classed principally by part of speech. Variations in the body of the text are produced by randomly selecting one of two sentence templates.



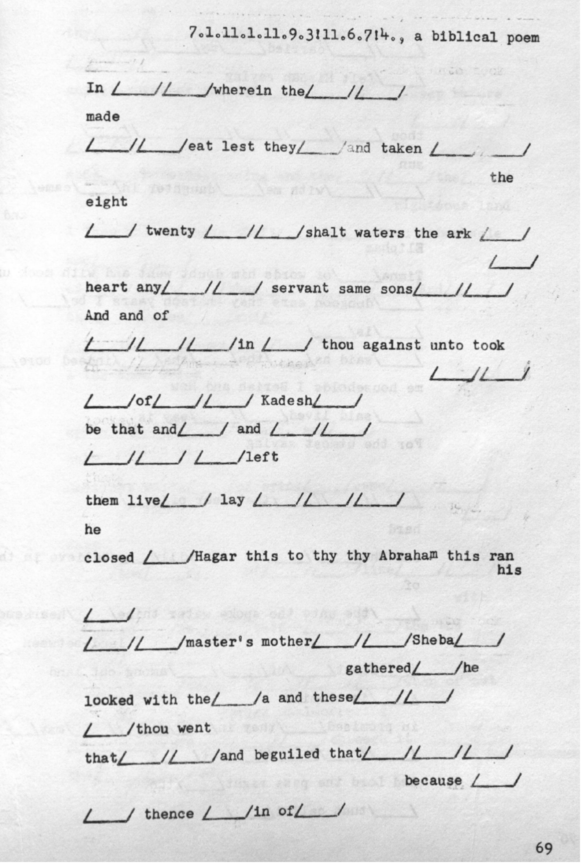
```
DUCK DUCK
   YOU ARE MY LITTLE AFFECTION:
MY BEAUTIFUL APPETITE: MY EAGER
HUNGER. MY COVETOUS LOVE LUSTS
FOR YOUR INFATUATION. MY YEARNING
ANXIOUSLY CLINGS TO YOUR FELLOW
FEELING.

               YOURS EAGERLY
                     M. U. C.
```

Loveletter program

Link, David (2006). “There Must Be an Angel. On the Beginnings of the Arithmetics of Rays.” *Variantology 2. On Deep Time Relations of Arts, Sciences and Technologies,* ed. Siegfried Zielinski and David Link. pp15–42. Cologne: König.

1955

- Jackson Mac Low (Fluxus poet from New York) used chance operations to generate five biblical poems.



```
                    7.1.11.1.11.9.3!11.6.7!4., a biblical poem

In /____//____/wherein the/____//____/
made
/____//____/eat lest they/____/and taken /____/,____/
                                                            the
eight
/____/ twenty /____//____/shalt waters the ark /____/
                                                       /____/
heart any/____//____/ servant same sons/____//____/
And and of
/____//____//____/in /____/ thou against unto took
                                                  /____//____/
/____/of/____//____/ Kadesh/____/
be that and/____/ and /____//____/
/____//____/ /____/left

them live/____/ lay /____//____//____/
he
closed /____/Hagar this to thy thy Abraham this ran
                                                      his

/____/
/____//____/master's mother/____//____/Sheba/____/
                                       gathered/____/he
looked with the/____/a and these/____//____/
/____/thou went
that/____//____/and beguiled that/____//____//____/
                                                because /____/
/____/ thence /____/in of/____/
                                                               69
```

Jackson Mac Low's early "biblical poems."

Acconci, Vito and Mayer, Bernadette (1968). *0 to 9* Magazine No. 4

1958

- "A high point of mechanical produced poems was reached with the text *Der Vogel singt. Eine Dichtungsmaschine in 571 Bestandteilen* (The Bird Sings. A poetry Machine in 571 Parts) by Konrad Bayer (based on a draft from Oswald Wiener), whose skeletal prose reveals a search for a reductionist form."

Weibel, Peter (2005). "Viennese Formal Film". *Beyond Art: A Third Culture: a Comparative Study in Cultures, Art, and Science in 20th Century Austria and Hungary.* Springer, pp. 142–143

1959

- Brion Gysin created 120 permutations of the biblical phrase I AM THAT I AM (programmed by Ian Sommerville).

I AM THAT I AM
AM I THAT I AM
I THAT AM I AM
THAT I AM I AM
AM THAT I I AM
THAT AM I I AM
I AM I THAT AM
AM I I THAT AM
I I AM THAT AM
I I AM THAT AM

"The poets are supposed to liberate the words – not to chain them in phrases. Who told poets they were supposed to think? Poets are meant to sing and to make words sing. Poets have no words 'of their own.' Writers don't own their words. Since when do words belong to anybody."

Gysin, Brion (1960). *Cut-Ups Self-Explained.* Available from: *homepages.spa.umn.edu/~duvernoi/cuts.html*

- Theo Lutz, *Stochastic text*

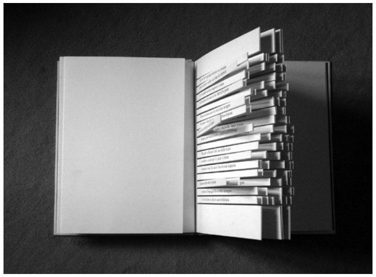
1960

- Foundation of the group *OuLiPo* (L'Ouvroir de Litérature Potentielle) by Raymond Queneau and the mathematician François Le Lionnais. Other members are for example Georges Perec, Italo Calvino, Oskar Pastior, Claude Berge and Jacques Roubaud. The group defines the term *littérature potentielle* and was active with the creation of new forms of literature and writing techniques, often based on mathematical principles. Georges Perec, for example, wrote a novel without the letter e (*La Disparition, 1969)*.

1961

- Nanni Balestrini (member of the *Group 63,* an avant-garde Italian literary movement of the 1960s) generated the Tape Mark poem. A computational collaged poem from 3 different writers: Lao Tzu (Tao Te Ching), Michihito Hachiya (Hiroshima Diary) and Paul Goldwin (The Mystery of the Elevator).

- Raymond Queneau published *Cent Mille Milliards de poèmes,* a book with ten sonnets whose pages are divided by cutting. Through recombination one could create 1 million poems. Published by Gallimard, Paris.



Cent Mille Milliards de poèmes

Picture available from: autrebistrotaccordion.blogspot.com/2012_03_01_archive.html

1962

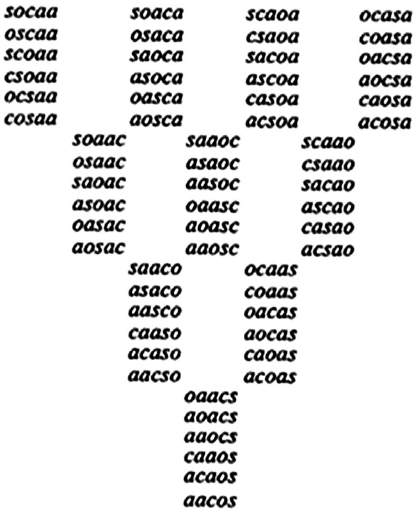
- François Le Lionnais wrote the first manifest for the group OuLiPo with the title *La Lipo.*

- R.M. Worthy (Laboratory for Automata Research) published a selection of poems that where generated with the computer program *Auto-Beatnik.* In: Horizon Magazine, May 1962.

- Douglas Engelbart, first hypertext system *Augment.* The system was inspired by Vannevar Bush's Memex. It was originally designed for mainframe environments, and was then available for workstations. Engelbart is the inventor of the computer mouse.

1963

- Augusto de Campos, member of the south-american group *Noigandres* produced ACASO, a permutation poem



socaa
oscaa
scoaa
csoaa
ocsaa
cosaa

soaca
osaca
saoca
asoca
oasca
aosca

scaoa
csaoa
sacoa
ascoa
casoa
acsoa

ocasa
coasa
oacsa
aocsa
caosa
acosa

soaac
osaac
saoac
asoac
oasac
aosac

saaoc
asaoc
aasoc
oaasc
aoasc
aaosc

scaao
csaao
sacao
ascao
casao
acsao

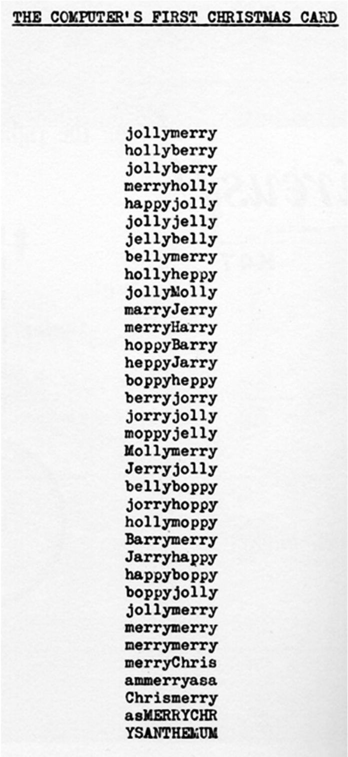
saaco
asaco
aasco
caaso
acaso
aacso

ocaas
coaas
oacas
aocas
caoas
acoas

oaacs
aoacs
aaocs
caaos
acaos
aacos

Picture available from: www.antoniomiranda.com.br/poesia_visual/augusto_de%20campos.html

- Edwin Morgan published *Computer's first Christmas Card.*



THE COMPUTER'S FIRST CHRISTMAS CARD

jollymerry
hollyberry
jollyberry
merryholly
happyjolly
jollyjelly
jellybelly
bellymerry
hollyheppy
jollyMolly
marryJerry
merryHarry
hoppyBarry
heppyJarry
boppyheppy
berryjorry
jorryjolly
moppyjelly
Mollymerry
Jerryjolly
bellyboppy
jorryhoppy
hollymoppy
Barrymerry
Jarryhappy
happyboppy
boppyjolly
jollymerry
merrymerry
merrymerry
merryChris
ammerryasa
Chrismerry
asMERRYCHR
YSANTHEMUM

Picture from: Mary Ellen Solt, *Concrete Poetry a World View,* 1968

- Clair Philippy produced blank verses with 150 words per minute (on a RCA 301 machine).

◆ Samuel R. Levin published *On Automatic Production of Poetic Sequences.*

1964

◆ Jean Baudot, a pioneering engineer-linguist, creates the first French machine-generated published poetry*. La Machine a écrire mise en marche et programmée par Jean A. Baudot.* The publication includes computer generated poems (program PHRASE), technical explanations and reactions of contemporary writers.

◆ Louis Couffignal and Albert Ducrocq create *Un doute agréable couleur de lotus endormi...,* an imitation surrealist poem created on Calliope hardware system.

◆ English critic Mike Weaver organized *The First International Exhibition of Concrete and Kinetic Poetry* in Cambridge

◆ François Le Lionnais (OuLiPo) hold a conference at the university of Liège with the title *Machines Logiques et Electroniques et Littérature.*

◆ Clair Philippy, computer scientists from the RCA Service Company, Pennsylvania, USA programmed a vocabulary of 100 words, which the computer arranged in verse.

◆ Max Bense and Reinhard Döhl (Stuttgarter Gruppe) wrote the manifest *Zur Lage*. It included a description of experimental texts.

Text available from: www.stuttgarter-schule.de/zur_lage.htm

1965

◆ Emmett Williams used the 101 most used words from *Dante's Divine Comedy* to create “Music”, a computer poem.

◆ Ted Nelson developed the idea of hypertext: a software that includes all kinds of possible literature (like the library of Babel) and allowes users to follow links to find further references.

1966

◆ Joseph Weizenbaum developed the interactive program ELIZA.

◆ Emmet Williams “The IBM Poem”

◆ Gerhard Stickel “Autopoeme” (Deutsches Rechenzentrum Darmstadt)

1967

◆ Jean A. Baudot creates a second version of his poem generator PHRASE (1964), REPHRASE, which is related to the play *Equation pour un homme actuel.*

- Manfred Krause and Götz F. Schaudt generate poems with the ZUSE Z 23. They published the book *Computerlyrik. Poesie aus dem Elektronenrechner.* (Droste)

- John Morris published the article *How to write Poems with a Computer* (Michigan Quarterly Review). He described a Haiku program, which used predefined vocabulary.

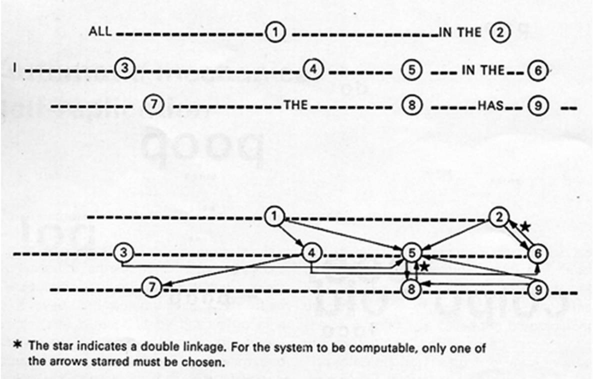
Text available from: dwwp.decontextualize.com/pdfs/morris.pdf (January 2012)

1968

- Emmet Williams (in cooperation with Peter G. Neumann) create the visual-sound-poem *Guillaume Apollinaire.*

- Jasia Reichardt curates the exposition *Cybernetic Serendipity* (London Institut for Contemporary Arts) which introduces computer generated poems.

- Computerized Haiku was shown in the exhibition *Cybernetic Serendipity* (ICA, London). Programmed by Margaret Masterman and Robin McKinnon-Wood.

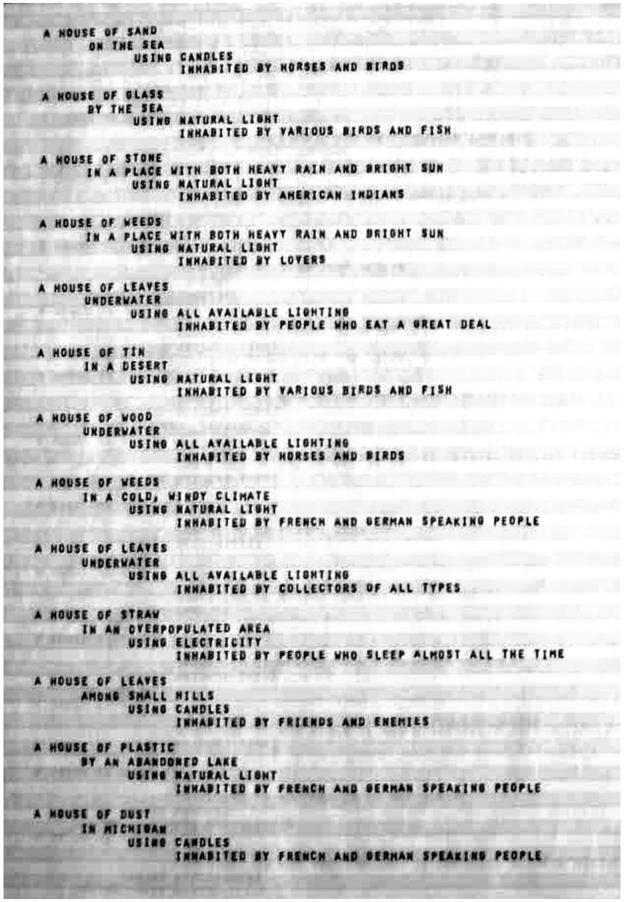


* The star indicates a double linkage. For the system to be computable, only one of the arrows starred must be chosen.

Computerized Japanese Haiku

Picture Available from: web.njit.edu/~funkhous/2008/machine/6.html

- *A House of Dust*, computer generated poem, creating stanzas by working through iterations of lines with changing words from a finite vocabulary list. A poetry project created by Alison Knowles and James Tenney



```
A HOUSE OF SAND
     ON THE SEA
          USING CANDLES
               INHABITED BY HORSES AND BIRDS

A HOUSE OF GLASS
     BY THE SEA
          USING NATURAL LIGHT
               INHABITED BY VARIOUS BIRDS AND FISH

A HOUSE OF STONE
     IN A PLACE WITH BOTH HEAVY RAIN AND BRIGHT SUN
          USING NATURAL LIGHT
               INHABITED BY AMERICAN INDIANS

A HOUSE OF WEEDS
     IN A PLACE WITH BOTH HEAVY RAIN AND BRIGHT SUN
          USING NATURAL LIGHT
               INHABITED BY LOVERS

A HOUSE OF LEAVES
     UNDERWATER
          USING ALL AVAILABLE LIGHTING
               INHABITED BY PEOPLE WHO EAT A GREAT DEAL

A HOUSE OF TIN
     IN A DESERT
          USING NATURAL LIGHT
               INHABITED BY VARIOUS BIRDS AND FISH

A HOUSE OF WOOD
     UNDERWATER
          USING ALL AVAILABLE LIGHTING
               INHABITED BY HORSES AND BIRDS

A HOUSE OF WEEDS
     IN A COLD, WINDY CLIMATE
          USING NATURAL LIGHT
               INHABITED BY FRENCH AND GERMAN SPEAKING PEOPLE

A HOUSE OF LEAVES
     UNDERWATER
          USING ALL AVAILABLE LIGHTING
               INHABITED BY COLLECTORS OF ALL TYPES

A HOUSE OF STRAW
     IN AN OVERPOPULATED AREA
          USING ELECTRICITY
               INHABITED BY PEOPLE WHO SLEEP ALMOST ALL THE TIME

A HOUSE OF LEAVES
     AMONG SMALL HILLS
          USING CANDLES
               INHABITED BY FRIENDS AND ENEMIES

A HOUSE OF PLASTIC
     BY AN ABANDONED LAKE
          USING NATURAL LIGHT
               INHABITED BY FRENCH AND GERMAN SPEAKING PEOPLE

A HOUSE OF DUST
     IN MICHIGAN
          USING CANDLES
               INHABITED BY FRENCH AND GERMAN SPEAKING PEOPLE
```

A House of Dust

Picture Available from: beineckeroom26.library.yale.edu/2010/02/15/a-house-of-dust/

- James Tenney, *Hank and Mary, a love story, a chorale*

1969

- Jackson Mac Low *PFR-3 Poems* (Art and Technology program, Los Angeles County Museum of Art).

1970

- Dick Higgins published the chapbook *Computers for the Arts,* a theoretical reflexion about computer poems.

- Terry Winograd published SHRDLU, a natural-language system that can understand typed commands and move blocks around on a program surface.

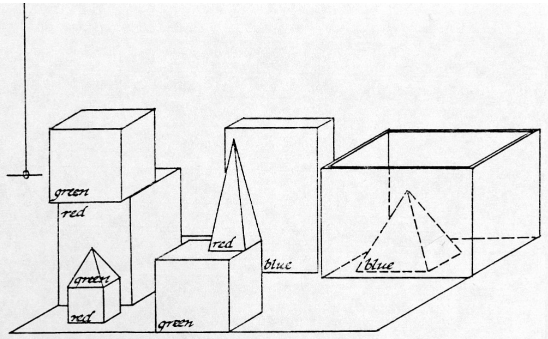


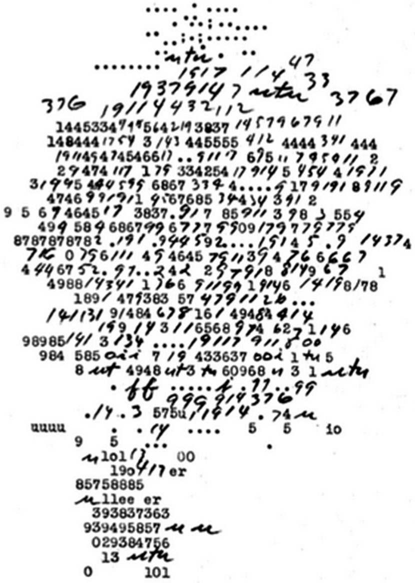
FIG. 3. "Pick up a big red block."

SHRDLU

Link, David (2002). *Poesiemaschinen/Maschinenpoesie.* Dissertation. Berlin: Humboldt University. Picture available from: edoc.hu-berlin.de

◆ Alan Sondheim, *4320*

◆ Carl Fernbach-Flarsheim, *The Boolean Image/ Conceptual Typewriter*



Conceptual Typewriter

Picture available from: modcult.org/read/2010/1/17/carl-fernbach-flarsheim

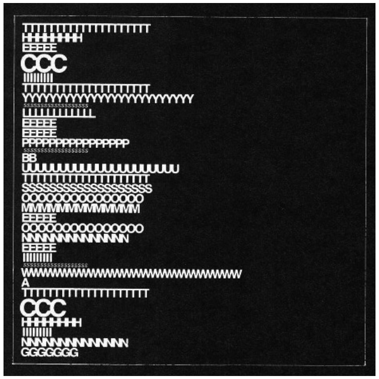
1971

◆ Amsterdam exhibition *klankteksten / ? konkrete poezie / visuele teksten*

◆ Gerrit Krol book *APPI: Automatic Poetry by Pointed Information,* Em. Querido's Uitgeverij N. V., Amsterdam

1972

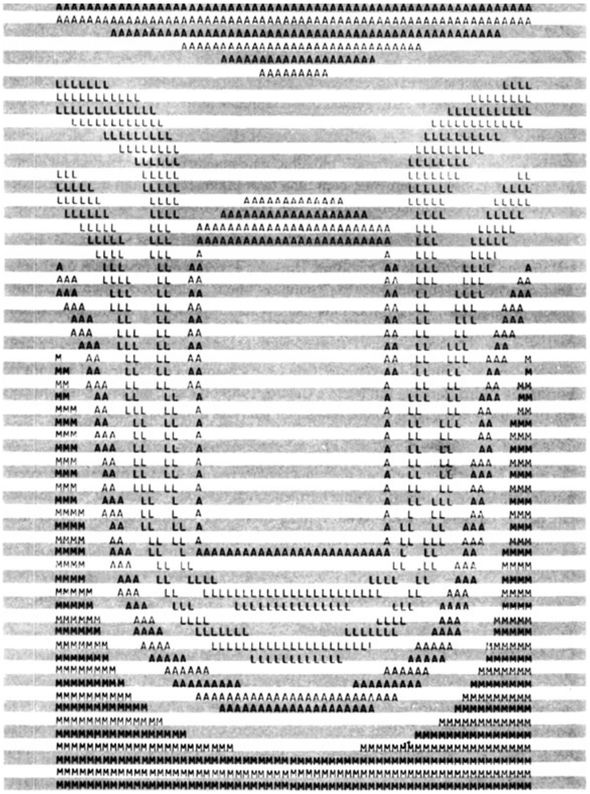
◆ Aaron Marcus, *The City Sleeps but Someone Is Watching*



The City Sleeps but Someone Is Watching

Picture available from: www.atari-archives.org/artist/sec4.php

◆ Erthos Albino de Souza, *Le tombeau de Mallarmé*



Le tombeau de Mallarmé

Picture available from: www.interact.com.pt/memory/15/images/stories/ed15/ed15_jla_8.jpg

1973

- Richard W. Bailey edits the *Computer Poems* anthology

- Kenneth Colby develops PARRY, an early chatterbot that mimics paranoid behaviour.

1974

- Ted Nelson publishes the book *Computer Lib / Dream Machines*

1976

- Ángel Carmona, *Poemas V2. Poesía compuesta por una computadora,* Producciones Editoriales

1978

- William S. Burroughs and Brion Gysin *The Third Mind.* They use the cut-up technique: taking texts, cutting the pages, and then rearranging the pieces to form a new narrative.

1979

- Phillippe Bootz published *La poésie matricielle*. He creates combinational poems with a mini-computer.

1980

- Jean-Pierre Balpe, automatic poetry generator *Poèmes d'Amour.*

1982

- Foundation of A.L.A.M.O. (Atelier de Littérature Assistée par la Mathématique et l'ordinateur) a workshop of mathematics and computer-assisted literature.

- Ted Nelson publishes the book *Literary Machines.*

1983

- John Cayley uses the HyperCard program for the production of *wine flying* to make the poetic structure of a classical Chinese quatrain visible.

1984

- William Chamberlain *The Policeman's Beard is Half Constructed* (written on the BASIC program RACTER, which was co-developed by Thomas Etter).

- bpNichol (Canadian poet) *First Screening*, computer poems, programmed in Basic for Apple II.

1985

- Jean-Francois Lyotard curates the exhibition *Les Immateriaux,* Paris, Centre Georges Pompidou, presenting the software MAOTH (text manipulation Assistée par de Ordinateurs hybrid) and LAPAL (Langage Algorithmique pour la production de Littérature Assistée).

- Stewart Brand founded the Bulletin Board Systems *Whole Earth Lectronic Link* (WELL) in San Francisco.

1986

- Philippe Bootz, telematic poems, *Metamorphose.*

- Jean-Pierre Balpe, event *La Bibliothèque du Futur* held at the Centre George Pompidou, Paris.

1987

- Judith Kerman, *Interactive Poem Demo Animated Picture Poems*

1988

- Jim Rosenberg publishes *Intergrams*

1989

Rod Willmot, *Everglade*, hypertext novel

1993

- Judith Malloy publishes the hypertext story *Its Name Was Penelope (*based on the Odyssey). The reader can select text passages, a pseudo-random number generator adds the decisions.

1994

- Foundation of the *Electronic Poetry Center* (SUNY Buffalo)

1995

- Alain Vuillemin and Michel Lenoble publish *Littérature et Informatique. La Littérature Génerée par Ordinateur.*

- Charles O. Hartman and Hugh Kenner, *Sentences.* Hartman took some didactic text and ran it through the TRAVESTY computer program developed by Hugh Kenner and Joseph O'Rourke. Hartman then ran the result through his own DIASTEXT program, based on the "diastic" writings of poet Jackson Mac Low. The result is a eccentric poetry similar to Surrealist automatic writing.

1998

- Hans Magnus Enzensberger, *Poesie-Automat.*

1999

- Giselle Beiguelman, *The Book after the Book*, an extended commentary upon online reading in the form of an online hypertext. Modelled after Jorge Luis Borges' "Book of Sand," The Book after the Book seeks an understanding of the ways in which hypertext and online reading change the ways we think about the book as volume, the library as place, and the relationship between image and text.

2000

- Peter Dittmer, *Die Amme.* Dialogues with a computer

- S. Bringsjord and D. A. Ferrucci, BRUTUS

2001

- David Link, *Poetry Machine 1.0.* The interactive installation operates with a keyboard as an interface, an Internet connection and two video screens. The word processor extracts associations from the World Wide Web linked to the input.

2002

- Christophe Bruno develops *Gogolchat.* Gogol takes words out of a chat and mixes them with found objects from the Internet.

Bibliography

Bense, Max (1969).“Automatentheorie der Texte”, “Synthetische Texte”. In: *Einführung in die informationstheoretische Ästhetik. Grundlagen und Anwedung in der Texttheorie.* Hamburg: Rowohlt Verlag.

Borges, Jorge Luis (2000 [1941]). *Library of Babel*. Boston: David R. Godine Publisher.

Bonacic, Vladimir (1971).“Art as function of subject, cognition and time”. In: *Boris Kelemen & Radoslav Putar: Dialog with the Machine*. Zagreb: Galerije Grada Zagreba.

Boden, Margaret A. (2004).“Creativity in a Nutshell”, In: *The Creative Mind: Myths and Mechanisms.* London: Routledge.

Brockman, John (2010). *The Edge Annual Question 2010: How Is the Internet Changing the Way You Think?* Available from: http://www.huffingtonpost.com/john-brockman/

Burroughs, William (1978 [1964]). J. Grauerholz, I. Silverberg (Ed.).“The Future of the Novel”. In: *The Word Virus.* New York: Grove Press.

Cope, Bill and Phillips, Angus (2006). *The Future of the Book in the Digital Age*. Oxford: Chandos Publishing.

Critical Art Ensemble (2001). *The Technology of Uselessness.* Available from: http://www.critical-art.net/

Dahl, Roald (1996 [1953]).“The Great Automatic Grammatizator”. In: *The Great Automatic Grammatizator and Other Stories*. Puffin Books: London.

Diodato, Virgil and Gellatly, Peter (1994). *Dictionary of Bibliometrics.* Haworth Press.

Dreher, Thomas. *Conceptual Art and Software Art: Notations, Algorithms and Codes.* IASL online NetArt: Theory. Available from: iasl.uni-muenchen.de/links/NAKSe.html (Nov. 2011)

Eco, Umberto (2003). *Vegetal and Mineral Memory: The Future of Books.*
Available from: www.bibalex.org/attachments/english/Vegetal_and_Mineral_Memory.pdf

Funkhouser, Chris (2007). *Prehistoric Digital Poetry: An Archaeology of Forms, 1959–1995.* University of Alabama Press.

Ioannidou, Ersi (2010). *Humanist Machines: Daniel Libeskind’s Three Lessons in Architecture*. Available from: Google Books.

Jeanneney, Jean-Noel (2006). *Google and the Myth of Universal Knowledge: A View from Europe.* Chicago: University Of Chicago Press.

Keenan, Stella (2000 [1996]). *Concise Dictionary of Library and Information Science*.
Martindale-Hubbell.

Kittler, Friedrich A. (1990 [1985]). *Discourse Networks 1800/ 1900.* Stanford: Stanford University Press.

Kittler, Friedrich A. (2008).“Vom Götterbild zur Computeranimation”. *Notation: Kalkül und Form in den Künsten.* Berlin: Akademie der Künste.

LeWitt, Sol (1999).“Paragraphs on Conceptual Art” *Artforum*, 5:10. Summer 1967, pp.79–84.
Reprinted in Alexander Alberro and Blake Stimson ed., *Conceptual Art: A Critical Anthology*, Cambridge: MIT Press. pp. 12–16.

Link, David (2002). *Poesiemaschinen / Maschinenpoesie.* Dissertation. Berlin: Humboldt University. Available from: edoc.hu-berlin.de

Packer, Randall and Jordan, Ken (Ed.) (2001). *Multimedia: From Wagner to Virtual Reality*. New York: Norton & Co. Available from: dichtung-digital.mewi.unibas.ch/index.htm, (Jan. 2011)

Parker, M. Philip (2007). *Method and Apparatus for Automated Authoring and Marketing.* US patent. Available from: patft.uspto.gov/, (Nov. 2011)

Pérez, Daniel et al. [no Date]. *Automatic Generation of Literary Texts: Greek Mythology.* Paper available from: data.bibbase.org

Pérez, Rafael and Sharples, Mike (2004). *Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA.* Paper available from: www.lsri.nottingham.ac.uk/msh/Papers/MEXICA%20KBS.pdf, (Nov. 2011)

RACTER (1984). *The Policeman's Beard is Half Constructed: Computer Prose and Poetry by RACTER – The First Book Ever Written by a Computer.* Warner Software/Warner Books.

Simanowski, Roberto (2001). "Text Machines". In: *Digital Art and Meaning.* Minneapolis: University of Minnesota Press.

De Sousa, Ronald (2000). Review of Selmer Bringsjord and David A. Ferrucci, "Artificial Intelligence and Literary Creativity: Inside the mind of BRUTUS, a storytelling machine." In: *Computational Linguistics.*

Stachowiak, Herbert (1965). *Denken und Erkennen im kybernetischen Modell.* Vienna/NY: Springer Verlag.

Stoicheff, Peter and Taylor, Andrew (2004). *The Future of the Page – Studies in Book and Print Culture*. Toronto: University of Toronto Press.

Sullivan, Nick (1997). *Regency Romance Writer.* Available from: www-ssrl.slac.stanford.edu/~winston/baers/romriter.html, (Nov. 2011)

Swift, Jonathan (1726 [1826]). *Travels into Several Remote Nations of the World.* London: Jones & Company. Available from: Google Books.

Weibel, Peter et al. (2008). *Notation. Kalkül und Form in den Künsten*. Berlin: Akademie der Künste.

Weibel, Peter et al. Curator. *Die Algorithmische Revolution*. Exposition, 2004, Zentrum für Kunst und Medientechnologie, Karlsruhe. Available from: www.zkm.de, (Nov. 2011)

Weizenbaum, Joseph (1966). "ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine". *Communications of the ACM*, Vol. 9, No 1. MIT: Cambridge, Mass.

Wittgenstein, Ludwig (2003 [1963]). *Tractatus logico-philosophicus.* Frankfurt am Main: Edition Suhrkamp.

Wiener, Norbert (1950). "Cybernetics in History". *The Human Use of Human Beings.* Da Capo Press. Reprint from Houghton Mifflin Company (Boston).

Image credits

FIG. 1 Sol LeWitt “Arcs and Lines” at Paula Cooper Gallery, 2011, New York, © Cathy Carver

FIG. 2 Sol LeWitt, “Wall Drawing #260”, MOMA, 2008, New York, © Jason Mandella

FIG. 3 Casay Reas, Available from: artport.whitney.org/commissions/softwarestructures/ (Jan. 2012)

FIG. 4 Plate V, Part III, Swift, Jonathan (1726 [1826]). *Travels into Several Remote Nations of the World.* London: Jones & Company. p. 74. Available from: Google Books.

FIG. 8–10 Racter (1984). *The Policeman‘s Beard is Half Constructed: Computer Prose and Poetry by Racter,* Available from: ubu.artmob.ca/text/racter/racter_policemansbeard.pdf (Jan. 2012)

FIG. 5 & 11 Reprint from: Link, David (2002). *Poesiemaschinen/Maschinenpoesie*. Berlin: Humboldt University. Available from: edoc.hu-berlin.de (Jan. 2012)

Cover *Available from:* www.psfk.com/2011/09/how-long-does-it-take-for-monkeys-to-reproduce-shakespeares-works.html#ixzz1teI9G6KD

Back side Eric Desmaziere, *Etchings For Borges’ Library Of Babel,* Available from: socks-studio.com/2011/05/01/eric-desmazieres-etchings-for-borges-library-of-babel/