Genetic Editions to the Extreme? Conserving Historical Text Generators

Claus-Michael Schlesinger
claus-michael.schlesinger@ilw.uni-stuttgart.de
Universität Stuttgart, Germany

Short Description

My talk will focus on the problems and possibilities of conserving historic text generators, 'historic' meaning roughly 1950-1970. For this purpose, I will develop the scientific interests in these generators either from the perspective of literary history and from the perspective of the history of knowledge, since many of the text generators in question were connected explicitly to scientific interests.

A system for conserving and editing historic text generators, that will enable researchers from both of these fields to access historic text generators in order to study their esthetics and functioning needs, in orderto take into account not only the generated texts, but also the generating texts, meaning software and prerequisites such as flowcharts. In my talk, I will focus on the materiality and the potentiality of the historic text generators in order to propose a platform solution that should enable scholars to edit (and publish) historic text generators.

Full Abstract

In their manifesto Zur Lage (Bense and Döhl (1964)) (State of Things) Max Bense and Reinhard Döhl name several tendencies of contemporary literature. The sixth and last point of their list calls on a cybernetic and material poetry ("kybernetische und materiale Poesie"). Historically and biographically, Bense and Döhl were placed at one of the centers of concrete poetry and concrete art (Stuttgart, Germany), and their manifesto reflects the artistic and academic developments made in these areas. But already in 1964, when Bense and Döhl published their manifesto, experiments had taken place that had taken the notion of a cybernetic poetry quite literally, using computers and software to generate literary texts. (Strachey Okt. 1954; Lutz 1959; Levin 1963; Gunzenhäuser 2004; Link 2004) The automatic generation of texts by way of combining syntax, vocabulary and a (pseudo) random generator that serves to fill the syntax positions with proper words from the vocabulary has a history dating back to the 17th century. However, it was the use of computers that prompted artists and researchers to expand their experiments and develop complex setups applying hundreds of syntactical structures and bigger vocabularies (as did, for example, Stickel [1967]).

The research on early text generators has focused on biographical data and the literary history of automatic text generation (Bülow 2007), the autoreflexivity of digital texts (Cramer 2003) and the genealogy of text generating systems (Link 2004). A literary and cultural history of text generators needs to take into account not only the texts that were produced, but also the modes and methods of production. In my talk, I will argue that in order to understand the history of automatic text generation, we need to join these different perspectives and research either the generated texts and the modes and methods of text generation.

Now the first problem when dealing with text generators (and the texts generated with them historically) is the availability of the material. Only a small fraction of historically generated texts has been published, and in most cases the documentation of the conceptual and technical setup is only partly available through publications, not to mention the actual code that was used to produce the text in the first place. This is due to the fact, that literary scholars have tended to ignore texts of this kind – that is, experimental text which barely fulfills any definition of poetry – and are only beginning to recognize them as a certain type of literature in its own kind. The second reason for a newly developed recognition for early text generation experiments is the fact that automatic text generation has become ubiquitous in our time, from simple twitter bots that employ the same methods as the early text generators (meaning simple syntactical structures filled with randomly chosen words), via the modern ELIZAs of contemporary electronic assistant systems, to complex machine learning algorithms that synthesize Shakespearean plays. (Goodwin 2016)

So in order to further research on early text generators, the multiple dimensions of text generators will have to be made available to researchers from all fields concerned with the history of automatic text generation. Since I am, by profession, a philologist, my approach is limited to the document side of things and does not consider hardware conservation or emulation.

Documents connected to text generators comprise flowcharts, punched tape, source code data on magnetic memory, print-outs, project documentation in various paper formats. For a digital representation, these materials can be digitized either in graphical or in textual format. Graphical representations can be annotated with their respective metadata describing their material, function, authorship etc. As for the texts contained in a text generator as a historic object, we are dealing with three different kinds of texts:

- texts that have been generated
- texts that have been used to generate texts (mainly software)
- texts that could have been or can potentially be generated (the full outcome of the underlying algorithm)

Texts in groups 1 and 2 are connected in a genetic way. If we take the Stochastic Texts by Theo Lutz and Rul Gunzenhäuser (Lutz 1959) as an example, we can see that there is a published version of the generated texts and a raw version (published in Büscher, 2004). The differences between these versions are significant. From a text-genetic perspective, these differences lead to the production of the texts or, in other words, the execution of the underlying program. The execution of the program leads to the implementation of the program or the source code, the source code leads to the conceptual implementation of the algorithm (e.g. flowcharts) and thus to the abstract underlying algorithm.

Thus, a genetic perspective on text generators leads not only to the raw version of the output of the computer, but to the implementation of the algorithm in source code as well. The title of my talk searches to reflect this notion of text genesis, because the link between source code and output is not a certain similarity between two texts, but a functional, algorithmic connection. I think that problematizing this kind of functional genetic connection can also be fruitful for philology dealing with texts where computers don't generate all of the text, but help in generating certain features of the text. In order to describe and show this kind of connection in different text generators, it will be necessary to develop standardized notions for annotation schemas (Currently, the annotation of source code is not part of the TEI Guidelines (TEI-Consortium and Lou Burnard 2014)

Finally, studying the poetics of a text generator might also mean to study more generated text than is (historically) available. A representation of a text generator thus not only should comprise the source code itself, but also the possibility to run the algorithm in order to generate some of the potential texts. The best solution would be, of course, to run the code on the historic machines it was developed for. Not only would this repeat the original experiments, it would also hopefully reproduce the glitches that can be seen in the raw output that has been preserved and archived. (Büscher 2004) However, such a procedure would be way too costly. An easier way is to implement the reconstructed algorithm in a modern programming language and to offer researchers the possibility to run the code without being forced to undertake major retrocomputing tasks just in order to gather some more textual outcome.

Since this is a work in progress, I am confident that by August 2017 I will be able to outline not only the problems, but also some solutions for implementing a platform and database to preserve and present historic text generators.

Bibliography

Büscher, B., von Herrmann, H.-C., Hoffmann, C. (2004). *Ästhetik Als Programm. Max Bense/Daten Und Streuungen.* Berlin: diaphanes.

Bense, M., and Döhl, R. (1964). *Zur Lage.*

Bülow, R. (2007). "SINN IST FERN - Wie Die Computer Dichten Lernten." In *Ex Machina - Frühe Computergrafik Bis 1979: Die Sammlungen Franke Und Weitere Stiftungen in Der Kunsthalle Bremen*, 134–72. München: Deutscher Kunstverlag.

Cramer, F. (2003). Exe.cut[up]able Statements. Das Drängen Des Codes an Die Nutzeroberflächen. Berlin.

Goodwin, R. (2016). "Adventures in Narrated Reality. New Forms & Interfaces for Written Language, Enabled by Machine Intelligence." March. https:// medium.com/artists-and-machine-intelligence/adventures-in-narrated-reality-6516ff395ba3#.5ko9ojqfq.

Gunzenhäuser, R. (2004). "Zur Synthese von Texten Mit Hilfe Programmges- teuerter Ziffernrechenanlagen." In *Ästhetik Als Programm. Max Bense/Daten Und Streuungen*, edited by Barbara Büscher and Hans-Christian von Herrmann Christoph Hoffmann, 170–83. Berlin: diaphanes.

Levin, S. R. (1963). "On Automatic Production of Poetic Sequences." *The University of Texas Studies in Literature and Language* V: 138–46.

Link, D. (2004). *Poesiemaschinen - Maschinenpoesie.* Berlin: Diss. Humboldt- Universität.

Lutz, T. (1959). "Stochastische Texte." *Augenblick* 4 (1): 3–9.

Stickel, G. (1967). "Monte-Carlo-Texte. Automatische Manipulation von Sprachlichen Einheiten." In *Kunst Aus Dem Computer*, edited by William E. Simmat, 53–57. Stuttgart: nadolski.

Strachey, C. O. (1954). "The 'Thinking' Machine." *Encounter* 3 (4): 25–31.

TEI-Consortium, Bauman, S., Burnard, L. (2014). *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* Version 2.6.0, 20.1.2014, Revision 12802. o. O.