

WEIZENBAUM JOURNAL OF THE DIGITAL SOCIETY
Volume 3 \ Issue 3 \ w3.3.2 \ 11-06-2023
ISSN 2748-5625 \ DOI 10.34669/WI.WJDS/3.3.2

Information on this journal and its funding can be found on its website:

<https://wjds.weizenbaum-institut.de>

This work is available open access and is licensed under Creative Commons Attribution 4.0 (CC BY 4.0):

<https://creativecommons.org/licenses/by/4.0/>

KEYWORDS

Weizenbaum

ELIZA

artificial intelligence

machine learning

chatbots

chatGPT

RESEARCH PAPER

The Limits of Computation

Joseph Weizenbaum and the ELIZA Chatbot

David M. Berry 

University of Sussex, d.m.berry@sussex.ac.uk

ABSTRACT

Developed in the 1960s by Joseph Weizenbaum, ELIZA is arguably among the most influential computer programs ever written. ELIZA – and especially its most famous persona DOCTOR – continues to inspire programmers, wider discussions about AI, and imitations. This original ancestor of all conversational interfaces and chatbots maintains a special fascination for engineers, historians, and philosophers of artificial intelligence (AI) and computing. With its ability to produce human-like responses using a relatively small amount of computer code, ELIZA has paved the way for a multitude of similar programs. These take the form of conversation agents and other human-computer interfaces that have inspired entire new fields of study within computer science. This paper examines Weizenbaum's contribution to AI and considers his more critical writings in the context of contemporary developments in generative AI, such as ChatGPT. Examining how ELIZA has been discussed can provide insights into current debates surrounding machine learning and AI.

1 Introduction

The promise of artificial intelligence (AI) is to capture and recreate the essence of humanity's most powerful capacities, namely, language, creativity, reasoning, and intelligence.¹ Progress is happening at breakneck speed, with the practical impact of AI almost entirely constrained to the past ten years, with marked acceleration since 2019. In 2022, breakthroughs such as OpenAI's ChatGPT and Stability AI's Stable Diffusion were seen to empower human creativity and productivity in language and art as never before. In this paper, I want to consider recent advances in so-called generative AI in relation to what many consider its precursor: ELIZA, a relatively simple chatbot (conversation agent) program that enabled a conversation-based interface within a computer.

Developed in the 1960s by Joseph Weizenbaum, ELIZA is arguably among the most influential computer programs ever written. ELIZA – and especially its most famous persona DOCTOR – continues to attract programmers, generate discussions, and inspire imitations. However, although it has had an impact on computer science and the culture more broadly, the original source code for ELIZA was never published or widely distributed.² Nonetheless, it represents the precursor to all chatbots, the progenitor of conversational human-computer interaction, and inspiration for popular imaginings of what computers could be – see, for example, HAL 9000 from the 1968 film *2001: A Space Odyssey* and the operating system in the 2013 film *Her*. This original ancestor of all conversational interfaces and chatbots maintains a special fascination for engineers, historians, and philosophers of AI and computing. Notably, to the extent that ELIZA has substantially influenced the history of computing, it has also forever entangled it with problematic assumptions of gender and class (see below). With its ability to produce human-like responses using a relatively small amount of computer code, ELIZA has paved the way for a multitude of similar programs. These take the form of conversation agents and other human-computer interfaces that have inspired entire new fields of study within computer science (Boden, 1977).

¹ The related area of Artificial General Intelligence (AGI) attempts to synthesize these elements into a single AI system.

² As part of an international research team formed in 2021, we have re-discovered the lost ELIZA source code in MAD/SLIP in the MIT archives. Details of this development have been published on the Elizagen archive website. <https://sites.google.com/view/elizagen-org/the-original-eliza>

Notably, in designing an appropriate interface for the complex Large Language Model (LLM) system GPT-3, OpenAI settled on the chatbot format when it launched ChatGPT in 2022.³ Although the company has yet to fully explain the logic behind using this text-based model, using the system makes apparent that it is probably the ability to create personas that pushed the developers to design the system as such. To explore this further, this paper makes the critical observation that it can be very helpful to look backward to understand the particularities of AI products such as ChatGPT and similar products. This means considering precursor systems, especially ELIZA. In particular, I want to examine the reflections presented by Joseph Weizenbaum (1976) in *Computer Power and Human Reason: From Judgement to Calculation*, which saw him frame his inquiries in relation to the irreducibility of human thought to logical functions and, in particular, to ELIZA.⁴ That book raises questions about the relationship between rationality and logic, especially the kind of logic that can be implemented within computer programs and mathematical formulations. As Weizenbaum (1976) argued, “the introduction of computers into our already technological society has [...] merely reinforced and amplified those antecedent pressures that have driven man to an ever more highly rationalistic view of his society and an ever more mechanistic image of himself” (p. 11). According to Weizenbaum, there should be limits to what computers ought to be tasked to do. This would mean establishing a normative limit to the deployment of computation due to the way that computers affect the desire of humans to find a place in the world. He worried about the pretense of sympathy or interpersonal respect when, strictly speaking, the computer followed an instrumental programmed logic, arguing that this deception meant that the software design was inadvertently transforming a potentially mutually transformative communicative encounter into an alienating one. He (1972) also foresaw the difficulty of contesting a computer’s decision, writing

[N]o human is any longer responsible for “what the machine says.” Thus there can be neither right nor wrong, no question of justice, no theory with which one can agree or disagree, and finally no basis on which one can challenge “what the machine says.” (p. 613)

It is important that we begin to understand these new AI systems, both in terms of their “back-end” operation and their “front-end” interfaces, because they are likely to become much more prevalent as computing systems. Much like James Watt’s 1776 steam engine, AI is often seen as general-purpose technology with many possible applications. As such, AI implementations are

³ Large Language Models (LLMs) use huge textual inputs to create generative AI systems which produce textual output. They have therefore come to known as Generative AI due to their capacity to produce seemingly creative or generative textual materials at scale. They are built on a type of machine-learning model called a Transformer that has an “attention mechanism” that accelerates its learning capacity and the quality of the textual output it can produce.

⁴ We might note here the gendering of computational systems, especially chatbots, whether ELIZA or contemporary interfaces such as Siri (for a fuller discussion, see Marino, 2006).

expected to have a transformational impact on the global economy. AI is already used in everyday contexts – consider, for example, email spam filtering, media recommendation systems, navigation apps, and payment transaction validation and verification. Despite the clear potential for vast impacts on society and individuals, mapping these out is a difficult task, particularly given the field of AI is often celebratory about its discoveries and the potential it holds.⁵ In regulatory terms, an AI system’s autonomy raises unique questions about liability, insurance, and fairness, as well as risk and safety and even ownership of creative content. These concerns require that we think carefully about transparency and bias, infrastructural decisions, and the kinds of new digital skills required to create, use, and critique them.

Our societies increasingly depend on digital technologies that incorporate computational – and, therefore, calculative – rationalities and that raise challenges around maintaining the sociological capacity for human reasoning (Berry, 2011). Our growing reliance on small software applications soon becomes problematic because they are automated, networked, and interconnected into larger software platforms and services, making their operation even more complex. Although many of these systems were initially designed to support or aid the judgment of people undertaking numerous activities, analyses, and decisions, their workings have long since surpassed the understanding of users, becoming, nonetheless, indispensable. In doing so, these devices transform the capacities for human reason by short-circuiting the convolutions of cognitive processes that are manifest in human reason and by privileging certain instrumental relations that manifest in logical processes. As such, “the feeling of powerlessness so ubiquitous among individuals in our society[...] the widespread alienation of people from one another and from their work[...] the perception of ordinary people that they are living in the interstices of a gigantic system” (Weizenbaum, as cited in Rosenberg, 1980, p. 49). These notions of persuasion, deception, and mediation that Weizenbaum invokes are extremely pertinent to the issues that prevail in the later trajectory of AI.

Notably, advanced capitalist societies experience economic anarchy interwoven with rationalization and technology in a manner that tends to discourage reflective labor. Under such conditions, the values of instrumental reason are accorded a privileged status because they are embodied in the concept of rationality itself. Confounding calculation with rational thinking implies that whatever cannot be reduced to a number is an illusion or metaphysics. Consequently, the conditions are created for a possible decline in the susceptibility of society to critical thinking, which also manifests in a weakening of the potential for individuation. The drive to use rationalization and the insertion of algorithmic ways of doing and thinking is extremely pronounced in our con-

⁵ To my mind, the fact that current machine learning and AI systems have the capacity to automate the “boring stuff” humans have had to do in contemporary digital infrastructures, applications, and processes is substantially more convincing—of the transformational potential of the technology—than its purported “creativity” or cultural automation.

temporary computational societies. The turn towards automated systems that use AI – or machine learning, as it is more accurately described – may contribute further to this alienating process. This shows the importance of explanatory understandings that can critique these systems and avoid falling back on behavioral abstractions that avoid well-founded accounts of their workings (Berry, 2021). As Weizenbaum argues, “incomprehensibility is not a necessary property of even huge computer systems. The secret of their comprehensibility lies in that these systems are models of very robust theories” (Weizenbaum as cited in Rosenberg, 1980, p. 45).

The extent to which our conception of rationality has been rearticulated in a form that makes human thought comprehensible as calculation rather than judgment is something that Hannah Arendt explored in relation to political decision-making and policy (1972, p. 11). This clearly influenced Weizenbaum’s thinking.⁶ Quoting Arendt, he wrote, “[T]hey [the makers and executors of policy] were not just intelligent, but prided themselves on being ‘rational’[... They] did not *judge*; they calculated[...] an utterly irrational confidence in the calculability of reality [became] the leitmotif of the decision making” (1976, p. 14, emphasis in original). Later in that same text, he elaborates:

[C]omputers can make judicial decisions, [and] computers can make psychiatric judgments. They can flip coins in much more sophisticated ways than can the most patient human being. The point is that they ought not be given such tasks. They may even be able to arrive at ‘correct’ decisions in some cases-but always and necessarily on bases no human being should be willing to accept. (p. 227)

To investigate these issues in relation to ELIZA, it is helpful to consider the context and some of the debates that Weizenbaum was concerned with as he designed and programmed the system. First, I want to examine the environment in which Weizenbaum was developing the ELIZA system to obtain a sense of how he conceptualized the interface’s affordances. Second, I intend to consider the similarities and differences between ELIZA and modern chatbots being developed using generative AI. I conclude by offering some reflections on this relationship and drawing implications from the warnings presented by Weizenbaum.

⁶ Weizenbaum mentions Mumford, Arendt, Ellul, Roszak, Comfort, and Boulding specifically as individuals expressing “grave concern about the conditions created by the unfettered march of science and technology” (Weizenbaum, 1976, p. 11). Weizenbaum notes in *Computer Power and Human Reason* that Lewis Mumford “read all of it” before publication (Weizenbaum, 1976, p. x).

2 ELIZA and Early AI

Joseph Weizenbaum was born to a Jewish family in Berlin on January 8, 1923, the son of master furrier Jechiel Weizenbaum and his wife Henriette.⁷ At the age of 13, he fled with his parents from Nazi Germany, emigrating to the United States, where the family lived in Detroit. He later said that he “had an introduction to the world in formative years of the miscarriage of the ultimate form of rationality” (Weizenbaum, as cited in Dembart, 1977, p. 1). Although he studied mathematics at Wayne University, in 1941, his “studies were interrupted by the war, during which he served in the military at the meteorological service of the Air Force” (Sack, 2018). He returned to university in 1946 and obtained his Bachelor’s in Mathematics in 1948 and a Master’s in 1950.⁸ After this, “he helped design and build a digital computer at Wayne University in Detroit” (MIT, 2008). In 1955, he moved to the West Coast to join General Electric to develop, in collaboration with SRI, a banking automation system called Electronic Recording Machine, Accounting (ERMA).⁹ ERMA was developed to help the Bank of America manage the huge numbers of paper checks that it was processing manually. In the early 1950s, the banking industry was essentially on the brink of a crisis:

[Between] 1943 and 1952, check use in the United States had doubled from four billion to eight billion checks written every year. Bankers projected by 1955 the number of checks would be increasing by approximately one billion per year, and, by 1960, 14 billion checks would be written each year. (Fisher & McKenney, 1993, p. 44).

The ERMA system allowed the use of magnetically encoded typefaces printed on the bottom of checks, enabling automated check processing using magnetic ink character recognition technology. When ERMA was finally demonstrated to the press, it was still not working fully, and a programmer was hidden in a back room to help hide glitches in the system, a technique known as “Wizard of Oz” programming, where a human stands in for an algorithm that is not presently working or complete. According to Fisher and McKenney (1993),

⁷ Weizenbaum died on March 5, 2008, in Germany.

⁸ Weizenbaum explained, [What prompted] me to go into mathematics[...] was that of all the things that one could study, mathematics seemed by far the easiest. Mathematics is a game. It is entirely abstract. Hidden behind that recognition that mathematics is the easiest is the corresponding recognition that real life is the hardest. That has been with me since childhood. (Weizenbaum, as cited in Dembart, 1977).

⁹ SRI International was founded as the Stanford Research Institute, a nonprofit corporation, by Stanford University in 1946. SRI became independent of the university in 1970, and changed its name to SRI International in 1977 as it became increasingly international in focus (SRI, 2016).

[The] SRI engineer[...] indicated with a thumbs-up signal that it was performing stably and the show could go on. This form of prototyping is considered beneficial at early stages of the design cycle as it provides a means of studying and understanding user expectations and requirements. Maudsley et al. (1993) argue that this approach is particularly suited to exploring design possibilities in systems [that] are demanding to implement. The ERMA demonstration went perfectly[,] and the press never suspected the [computer system] was less than ideal. (p. 55)

Rheingold (1985) described the successful eventual computerization of this process, as a “milestone in the computerization of the world’s banking system” (p. 164). However, this early use of deception in computing to stand in for and smooth the introduction and use of algorithms – together with the replacement of human labor with computers – foreshadows the concerns Weizenbaum later raised about automation.¹⁰

While at General Electric, Weizenbaum began developing a programming system called Symmetric List Processor (SLIP). He completed it in 1963. SLIP was a set of functions for performing list processing by calling routines that was originally written in machine code and later in the Fortran programming language. Weizenbaum himself referred to it as a “FORTRAN/SLIP program” (1963, p. 524). Before leaving General Electric, Weizenbaum wrote a revealing article entitled “How To Make a Computer Appear Intelligent,” quoting Marvin Minsky to argue that an activity that “produces results in a way [that] does not appear understandable to a particular observer will appear to that observer to be somehow intelligent, or at least intelligently motivated” (Minsky, as cited in Weizenbaum 1962, p. 24). That paper also sees Weizenbaum describe a “five-in-a-row” game that gives the appearance of intelligence despite its underlying programming not actually encoding such intelligence:

[T]he author of an “artificially intelligent” program is, by the above reasoning, clearly setting out to fool some observers for some time. His success can be measured by the percentage of the exposed observers who have been fooled multiplied by the length of time they have failed to catch on. (1962, p. 24).

¹⁰ It is not clear whether Weizenbaum was present for the ERMA launch given that he joined the company in 1955 or 1956. Nonetheless, Weizenbaum is not listed among the principal engineers or contributors by Fisher and McKenney (1993), although it is likely that he would have heard about the ruse of disguising the operation of the computer by placing an engineer between the computer and the demonstrator.

In 1993, reflecting on his work, he later admitted that this established his reputation as a “charlatan or con man” (Weizenbaum, as cited in Crevier, 1993, p. 133).¹¹ He argued that, “in a way, that was a forerunner to my later ELIZA, to establish my status as a charlatan or con man. But the other side of the coin is that I freely stated it. The idea was to create the powerful illusion that the computer was intelligent. I went to considerable trouble in the paper to explain that there wasn’t much behind the scenes, that the machine wasn’t thinking. I explained the strategy well enough that anybody could write that program, which is the same thing I did with ELIZA”. (Weizenbaum, as cited in Crevier, 1993, p. 133).

On the basis of his work on SLIP he was offered an associate professorship in electrical engineering at Massachusetts Institute of Technology (MIT) in 1963. Within four years, he had been awarded tenure and a full professorship in computer science and engineering (in 1970).¹² There, he rewrote SLIP in MAD (Michigan Algorithm Decoder), a programming language and compiler developed in 1959 for the IBM 7090 and related computers. Although MAD was based on the ALGOL language, it is not an ALGOL language as such. The term MAD-SLIP is often used to designate the combination, but really, SLIP was always separate from MAD, representing a set of list-processing functions that could be incorporated into a MAD program. This counters a persistent misconception that ELIZA was originally written in LISP and not MAD-SLIP. Notably, Weizenbaum later confirmed that ELIZA had grown out of the game he wrote in 1962.¹³ According to Crevier (1993),

¹¹ One wonders whether Weizenbaum had read David Maurer’s *The Big Con*, published in 1940. Maurer later wrote, “[N]ow, confidence men do not go about burdened down with bales of script written out in advance to take care of every situation which may develop. They do not write out anything. But they know from experience the situations which are likely to come up, and know their lines letter-perfect for those situations. Then there are a number of stock variations which can be used if the office is given by the insideman. All the players know these variations by rote, and can swing into their routine at the given signal” (Maurer, 1968, p. 161; for a similar discussion, see Weil 2015). A better description of ELIZA is hard to imagine. Weizenbaum, interestingly, also includes a short digression about the “confidence man” in *Computer Power and Human Reason: From Judgement to Calculation* (1976, p. 121).

¹² Weizenbaum later held academic appointments at Harvard University’s Graduate School of Education, Stanford University, the Technical University of Berlin, and the University of Hamburg in Germany. He was a fellow of the American Association for the Advancement of Science, a member of the New York Academy of Science and of the European Academy of Science (MIT, 2008).

¹³ The game Weizenbaum (1962) described was actually five-in-a-row, rather than checkers, as suggested but not cited in the context of “machine learning” in the work of Samuel (1959). Nonetheless, Weizenbaum does reference Shannon’s papers on chess (1950; 1955) and would have likely been very influenced by Samuel’s 1959 paper.

In the mid-1960s at MIT, Weizenbaum was trying to get computers to talk to people in English. He noted that the programs in existence, like STUDENT, had only limited domains of application. Further, the knowledge describing these domains was inextricably bound to the program structure itself. Weizenbaum also deplored the limited capacity of the programs to acquire more information from the users by asking them questions. “I was smart enough to know that I couldn’t solve that in the next few weeks,” Weizenbaum told me, “so I started thinking about alternatives[...] I took all my tricks, put them in a bundle, and started this ELIZA business.” (pp. 133–134)

ELIZA attempted to simulate, in a relatively simple way, the conversation one might have with a psychotherapist. It used an interactive interface that enabled the user to type answers to questions generated by the software. Famously, many people were impressed by its ability to engage in conversation, assuming that the program possessed a greater amount of understanding and intelligence than was actually the case. After writing ELIZA, Weizenbaum admitted that “what I had not realized [was] that extremely short exposures to a relatively simple computer program could induce powerful delusional thinking in quite normal people” (1967, p. 7). As he explained, “this mode of conversation was chosen because the psychiatric interview is one of the few examples of categorized dyadic natural language communication in which one of the participating pair is free to assume the pose of knowing almost nothing of the real world” (1966, p. 42; see also Boden, 1977, p. 108).¹⁴ Using relatively simple pattern-matching techniques, Weizenbaum programmed ELIZA to transform inputs of English sentences into outputs that could appear to make sense to the user. According to McCorduck (2004),

[Q]uestion-answering machines were in the air [at MIT]. Bobrow was at MIT working on STUDENT; Raphael was working on what would be SIR; the Baseball program was a Cambridge-area product. To add impetus, Weizenbaum drove into work many a morning with his neighbor Victor Yngve, who had developed the COMIT language, for pattern matching. (p. 292)

Weizenbaum explored many of these issues in the ELIZA program, which was named after Eliza Doolittle, a working-class character in George Bernard Shaw’s 1912 play *Pygmalion*. Weizenbaum later actually admitted he came across the character from a casual encounter with an adaptation of the play into a musical called *My Fair Lady*, which debuted in 1956 and was followed by a film version released in 1964. His name for the script that created the persona that later became conflated with ELIZA was DOCTOR. Although that later may have been a gender-neutral name for a psychotherapist, ELIZA bears

¹⁴ Interestingly, in an interview, Weizenbaum stated that he “originally conceived of ELIZA as a barman, but later decided psychiatrists were more interesting” (Crevier, 1993, p. 136).

the legacy of its namesake, infusing a tale of class and gender and even racial and ethnic social formation into the program as the namely of ELIZA takes on the assumptions built into representations of women and social class both in the Shaw play, but also in the 1960s musical adaptation.

ELIZA responded to any English input with a typed response, and its script DOCTOR aimed to simulate or “parody” a Rogerian psychotherapist. The Rogerian approach to psychotherapy was developed by psychologist Carl Rogers in the 1940s and relied on a non-directive linguistic approach where the client is encouraged to do most of the talking. Weizenbaum used this as a model for structuring ELIZA’s responses which were mostly questions that repeated back the key elements of the user’s previous textual input in the conversation. Weizenbaum (1967) described ELIZA as the “first program [of]... a particular member of a family of programs which has come to be known as DOCTOR. The family name of these programs is ELIZA“ (p. 474). This set of programs was created in MAD-SLIP on an IBM 7094 computer at MIT, and published only as a general description (Weizenbaum, 1966; Weizenbaum, 1967). Describing how sentences were parsed, Weizenbaum (1966) wrote,

an input sentence is scanned from left to right. Each word is looked up in a dictionary of keywords. If a word is identified as a keyword, then (apart from the issue of precedence of keywords) only decomposition rules containing that keyword need to be tried. (p. 38)

He then moved on to detail how certain keywords required unique treatment: [I]t is very often true that when a person speaks in terms of universals such as “everybody,” “always[,]” and “nobody[,]” he is really referring to some quite specific event or person. By giving “everybody” a higher rank than “I,” the response “Who in particular are you thinking of” may be generated. (Weizenbaum, 1966, p. 39)

Weizenbaum also provided a basic flowchart illustrating keyword detection, and described several specific algorithms for weighing keywords:

Weizenbaum decided that the domain knowledge would reside in a program module separate from the one handling the conversations. He reasoned that if different kinds of knowledge were described in different knowledge modules (or “scripts,” as he called them), the program could then chat about a variety of topics. Feel like talking about haute couture rather than baseball today? Just load the haute-couture software module! Since the program could learn increasingly better speech, like Eliza Doolittle in George Bernard Shaw’s *Pygmalion*, Weizenbaum named it after her. (Another reason, as Weizenbaum later pointed out, was that, “like Miss Doolittle, it was never quite clear whether or not [the program] became smarter.”) (Crevier, 1993, p. 134).

3 Social Implications of ELIZA

By developing the program ELIZA and the DOCTOR script, Weizenbaum revealed that computation in its relationship to human behavior and reason would have profound effects, which he attempted to explore, contest, and draw limitations on:

I was startled to see how quickly and how very deeply people conversing with the DOCTOR became emotionally involved with the computer and how unequivocally they anthropomorphized it... [This was] clear evidence that people were conversing with the computer as if it were a person who could be appropriately and usefully addressed in intimate terms. (Weizenbaum, 1976, p. 6).

Indeed, the tendency to invest private feelings in a computer puzzled and concerned Weizenbaum, who worried that people's internal reality might be replaced by that of the machine. Weizenbaum was also concerned by the extent to which computers "induce powerful delusional thinking in quite normal people" and strengthened notions of human beings as machines, by which rationality became associated with calculation. This became known as the "ELIZA effect," the propensity for humans to ascribe understanding and intelligence to computer systems. Hofstadter (1995, p. 167) described it as "the susceptibility of people to read far more understanding than is warranted into strings of symbols – especially words – strung together by computers," a compelling description written in 1995 that, nonetheless, accurately describes modern generative AI systems (e.g., ChatGPT).

Here, we need to keep in mind that Weizenbaum's version of DOCTOR was printed more or less permanently on paper, rather than transiently appearing on the screen, [which might] influence how we understand the system as literary and as psychotherapeutic. A session that leaves a printed record, like a diary, may be experienced differently than a transient on-screen encounter. (Montfort, 2004)

Indeed, today, we are more familiar with ELIZA on screens, whether a desktop computer, a laptop, a smartphone, or a tablet. But computer systems in the 1960s almost always printed the output onto paper using a teletype machine. This absence of a screen is crucial for appreciating the material specificity of computation at this time and what it would have been like to experience using the system. Nonetheless, it is clear that ELIZA captured a mode of thought and an anxiety about our relationships with computers that were later explored by Weizenbaum (1976) through a critical examination of the philosophical, social and political grounds on which he understood his work. We should also note that software is a dynamic and changing medium. There are rarely single versions of a software program, indeed there are often multiple and contradictory copies created during programming.

Notably, there were multiple versions of ELIZA – Weizenbaum admitted that he continued work on ELIZA because “difficulty with the [original ELIZA was] that it [could] do very little other than generate plausible responses.” Later versions, he wrote (1967), differ from the old one in two main respects. First, it contains an evaluator capable of accepting expressions (programs) of unlimited complexity and evaluating (executing) them. It is, of course, also capable of storing the results of such evaluations for subsequent retrieval and use. Secondly, the idea of the script has been generalized so that now it is possible for the program to contain three different scripts simultaneously and to fetch new scripts from among an unlimited supply stored on a disk storage unit, intercommunication among coexisting scripts is also possible. (p. 478)

Sadly, these later advances were poorly documented and much of the original source code remains missing. Nonetheless, source code listing of ELIZA and some of the scripts from 1966 has now been recovered from the MIT archives.¹⁵

As part of a project I am involved in that seeks to reconstruct the ELIZA code, we have been able to ascertain that there were actually at least five major versions of ELIZA, three of which have been forgotten (and possibly lost) and a final planned version of which there are only textual references in other documentation. In chronological order, we have tentatively identified the following versions:

- 1) **ELIZA 1965a:** Delimits sentences with only “.” and “;” (evidence from MIT archive flowcharts);
- 2) **ELIZA 1965b:** Delimits sentences with “.” and “;” and “but”. Lacks the NEWKEY function. Includes undocumented CHANGE function, and hardcoded messages (the version recovered from the MIT archives);
- 3) **ELIZA 1966 CACM:** Includes the NEWKEY function, keyword stack and “but” delimiter.
- 4) **ELIZA 1967:** Adds sophisticated script handling. Evidence from not only descriptions in Weizenbaum (1967) and Taylor (1968) but also the extant ARITHM, F29, FIGURE scripts in the archive

¹⁵ The scripts, in addition to DOCTOR, which have been found in the MIT archives include: ARITHM (undertakes mathematical calculations within the conversation and is able to evaluate and return the result), F29 (appears to be an early version of the FIGURE mathematical definition script), FIGURE (appears to be F29 extended with many more definitions and responses), GIRL (which is a simple demonstration script), NEWENG (which discusses New England states). Scripts we have the output for, but not the script itself include: ELEVTR (discusses the physics of an elevator), POLETA (which discusses the “Pole and Barn Paradox”), SPACKS (discussion of a quoted line of poetry by Barry Spacks of the MIT humanities department), SYNCTA (the name of the script that discusses time synchronization). There are also references to scripts that we have neither the script nor its output: ANTIPR, INTRVW (described as an interview preliminary to the study of 4-vectors), FVP1, FVQUIZ, CANVEC, FRANCE, FORVEC, MIT, ORTH1, PHOTON, RLPOLN, STATES, QMPROB, WATSNU, XYPOLN. Work on the reconstruction can be found at <https://wg.criticalcodestudies.com/index.php?p=/discussion/108/the-original-eliza-in-mad-slip-2022-code-critique>

- 5) **ELIZA 1968+**: A description of the planning of this version appears in Taylor (1968). However, the scripts SPACKS, INTRVW, and FVP1 also give evidence of its use in much more sophisticated programming through the text of the scripts developed by Walter E. Daniel

Weizenbaum thought that ELIZA pointed to examples of human reliance on and trust in computers that represented an element of a much larger problem, namely,

[S]cience promised [humans] power. But, as so often happens when people are seduced by promises of power, the price exacted in advance and all along the path, and the price actually paid, is servitude and impotence[... But] power is nothing if it is not the power to choose. Instrumental reason can make decisions, but there is all the difference between deciding and choosing” (Weizenbaum, 1976, p. 259).

At the time, there were contentious debates around ELIZA and the development of PARRY, a competing system designed by Kenneth Colby in 1972 (for an exemplary combination of the two, see Cerf, 1973). Colby’s attempt to create a chatbot with paranoid schizophrenia was very controversial at the time and drove wider interest in the implications of the software. These debates continue to inform contemporary discussions around chatbots and AI today.

The issues that ELIZA raised were more broadly shared in popular culture. For example, its possibilities were fictionalized in The Firesign Theater’s 1971 album, *I Think We’re All Bozos on This Bus*. On this album, the protagonist Clem visits “The Future Fair,” where he talks to a computer that answers visitors’ questions with vague, positive-sounding replies only remotely related to the questions asked. When Clem – his name misheard by the fair’s personalization system as “AhClem” – has his opportunity to ask a question, he instead decides to hack it, accessing the system’s maintenance mode by saying, “This is worker speaking. Hello..” The “DOCTOR MEMORY” computer responds with “Systat: uptime” and the length of time that it has been running. Clem then attempts to crash the system by confusing it with questions that it cannot understand or, sometimes, even parse. Eventually, Clem asks, “Why does the Porridge Bird lay its egg in the air?” This causes the computer to put itself out of service and shut down. The remarkable appearance of a quasi-ELIZA system on this recording prompted many who recognized the jargon to wonder how esoteric technical jargon ended up on a comedy album. This remained a mystery until 2015, when Phillip Proctor, an original member of The Firesign Theatre, used a Quora comment to explain: “I got all my computer language [in the episode] from a printout of people with the ELIZA interactive psychiatrist program I found at a Work Fair in Los Angeles” (Proctor, 2015). The pop culture moment was later referenced by Apple iPhone’s AI chatbot, Siri, which would respond to the query, “This is worker speaking. Hello,” with “Hello Ah-Clem. What

function can I perform for you?”¹⁶ The story of how this came to be included in Siri, according to Proctor, was that “Steve Jobs, who he met at a Pixar screening of a film in which Proctor was a voice actor, told him that he was ‘a big Firesign fan’ and insisted this be include[ed] in Siri” (Tannenbaum, 2015).¹⁷ As shown here, in these examples ELIZA’s influence and the issues it raised were translated from the technical sphere into the cultural sphere and added to a wider public unease with the emergence of intelligent machines.¹⁸

Shortly after the publication of Weizenbaum’s 1966 paper, Bernie Cosell programmed an ELIZA in LISP (1966, 1969, 1972). Although based on Weizenbaum’s published algorithm, Cosell crucially had not seen the actual MAD-SLIP source code. LISP was rapidly becoming the primary language of AI, and Weizenbaum’s DOCTOR script is formatted in exactly the manner of a LISP symbolic expression. These factors probably cemented the persistent misconception that ELIZA was originally written in LISP, as did its circulation in small academic circles. Notably, Cosell’s code was also not widely published and was itself only recovered in 2013. The most famous Cosell version is considered to be the 1966 version, on the basis of which the GNU EMACS version and Jeff Shrager’s BASIC version were written, the latter first in 1973 and then as a reprint in a 1977 edition of the *Creative Computing magazine*.¹⁹ Shrager’s ELIZA was short enough to be quickly re-entered by hand, and would run on any personal computer with BASIC, which was very common at the time. (Shrager, 2015).²⁰ Similar programs exist for many systems today, including mobile phone versions and a Java version that can be run in web browsers. These versions differ slightly in that they run in different programming languages and differ in length – Cosell’s LISP version comprised 2,500 lines of code, while Shrager’s BASIC program is only 250 lines, with other versions falling between the two.²¹

¹⁶ Sadly, this easter egg appears to have been removed at some point in the last few years.

¹⁷ The recent turn towards conversational interfaces, also known as audible interfaces, replicates many of ELIZA’s features and can be seen in John Cayley’s *The Listeners* (2015–16), a linguistic performance, installation, and Amazon-distributed third-party app “transacted between speakers or speaker-visitors and an Amazon Echo.”

¹⁸ More recently, ELIZA appeared in Adrian Tchaikovsky’s 2015 science fiction novel, *Children of Time*.

¹⁹ Richard Stallman has long been considered the author of the GNU EMACS version of ELIZA but Stallman informed me that he had not written it (personal correspondence, November 13, 2022).

²⁰ DOCTOR was Bernie Cosell’s LISP conversion of ELIZA based on Weizenbaum’s description in his journal article. Bernie Cosell was one of a team of programmers at Bolt Beranek and Newman (BBN) who created the two specialized computers – called Interface Message Processors (IMPs) – that routed traffic on ARPANET in 1969. In September 1972, Vint Cerf documented a recording of a conversation between DOCTOR and PARRY, with the resulting conversation recorded as a Request for Comment in 1973 (Cerf, 1973).

²¹ I would like to thank participants in the Critical Code Studies Working Group 2016 and 2022 for rich and detailed discussions of ELIZA that substantially contributed to my thinking about ELIZA and DOCTOR.

Weizenbaum later went on to write about the dangers of manipulation and misrecognition in *Computer Power and Human Reason* (1976), which saw him argue that computers could have unforeseen negative impacts on human users. His early experiences automating human work, such as replacing bank clerks with computing machinery, informed Weizenbaum's future understanding of the challenges of computation to society. Weizenbaum was particularly deeply concerned about the substitution of human reason for effective procedures. For Weizenbaum, an effective procedure is an operation that enables the definition of a problem domain within formal mathematics, leaving it open for later computation using a machine. This formalization of thought creates the conditions under which aspects of human life are delegated as machine processes (e.g. using computation as the exemplar for thinking), introducing the problem that most people

don't understand computers to even the slightest degree. So unless they are capable of very great skepticism[... T]hey can explain the computer's intellectual feats only by bringing to bear the single analogy available to them, that is, their model of their own capacity to think. (Weizenbaum, 1976, p. 10)

That is, there is a common assumption that a behavioral abstraction of the computer helps to explain the operation of the machine. The drive to use rationalization and the insertion of algorithmic ways of doing and thinking is extremely pronounced in our contemporary computational societies and has only accelerated since the time of Weizenbaum's original writings. Examples include the introduction of measurable indicators of performance and standards of output and the monitoring and surveillance that computation makes possible. We also see the expansion of routine work in contrast to (dialogical) interaction in a capitalist society. However, equally seriously, there is a lack of legitimacy associated with algorithmic systems, partly because they remain opaque while contributing to the structural problems associated with democratic authority, including their increasing deployment in communications and media systems, with their accelerating influence suggesting the potential to generate systemic crisis and dangerous system failures (Berry, 2014). We might only reflect on the Great Financial Crisis (2008–2011) to see how calculative reason combined with rationalization and computation can create a heady mix in relation to profit-oriented corporations and individuals. Weizenbaum was keenly aware of these problems and sought to articulate the important issues at stake, observing that “our society's growing reliance on computer systems that we initially intended to ‘help’ people make analyses and decisions[...] has long since surpassed the understanding of their users and become indispensable to them[...] This] is a very serious development” (Weizenbaum, 1976, p. 236).

Weizenbaum gestures to the processes of reification and the increase in formalization made possible by algorithmization and the ways that we create systems where no one any longer knows explicitly or understands how they work or how they make decisions. This can create the danger that we become reluctant to modify them for fear of the unknown consequences. This raises the question of the creation and maintenance of people's intersubjective, historical concepts as manifested in and supported by augmenting technologies. This is captured by Weizenbaum's 1976 observation that "*The New York Times* has already begun to build a data bank of current events [... H]ow long before what counts as facts is determined by the system, before all other knowledge, all memory, is simply declared illegitimate?" (p. 238). Already, the ability to mediate between complex systems, vast data collections, and fast-moving data streams conditions society for the use of "scale" as a new intellectual horizon. Across industry and intellectual inquiry, we see a growing use of computational systems to abstract, simplify, and visualize complex "Big Data" phenomena and a tendency to use simplistic causal and statistical models to understand complex social phenomena, such as the notion of "social physics" (Pentland, 2015). Weizenbaum would observe that this mixture of "technological and political and social inevitability is a powerful tranquilizer of the conscience" that gives computation the power to not only decide but also act (Weizenbaum, 1976, p. 241).

Weizenbaum drew from his experience with ELIZA and other systems to argue that we must remain critical of AI and its technical development to both understand the contemporary situation and to prevent it spiraling out of our control:

I don't quite know whether it is especially computer science or its sub-discipline [AI] that has such an enormous affection for euphemism. We speak so spectacularly and so readily of computer systems that understand, that see, decide, make judgments, and so on, without ourselves recognizing our own superficiality and immeasurable naivete with respect to these concepts. And, in the process of so speaking, we anesthetize our ability to evaluate the quality of our work and, what is more important, to identify and become conscious of its end use. (Weizenbaum, 1987, p. 44)

4 Modern Iterations

The dangers from the automation of human thought and labour that Weizenbaum pointed towards calls for an approach to explanation that refuses to ignore and smooth over contradictions and contradictory claims at the phenomenological level and that attempts to grasp the dynamic moment of the subject. By leaving open the possibility of a critical reflexive understanding of algorithmic history and tradition, such an approach accepts the importance of the meaning structure of tradition but also seeks to avoid idealizing it. That is, in these computational societies, traditions might continue to embody interaction based on deception and distortion (i.e., ideology). This can often be translated, unreflexively, into algorithmic forms, echoing the words of Weizenbaum:

[T]he various systems and programs we have been discussing share some very significant characteristics: they are all, in a certain sense, simple; they all distort and abuse language; and they all, while disclaiming normative content, advocate an authoritarianism based on expertise. (Weizenbaum, 1976, p. 248)

To further emphasize this, I now want to look briefly at an example that illuminates how the interface possibilities of ELIZA and our expectations of how computers function can interact in ways that are profoundly anti-human. In this case, social conflict is embedded within the machinery of algorithms and labor is transformed into a commodity through the interface. This is an example of computation serving to hide social labor, such that workers are hidden “behind web forms, [chatbots] and APIs [that help] employers see themselves as builders of innovative technologies, rather than employers unconcerned with working conditions” (Irani & Silberman, 2013: 613). In this context, the “freedom” and “creativity” made possible by algorithms might actually be disguising systems of control and management.

To understand the ways that social conflict is submerged within the algorithmic form, it is helpful to investigate the modern generative AI systems (or chatbots) that new computational technologies make possible. In a sense, this is the “cooperation between brains” that Stiegler (2010, p. 47) argues is “produced through grammatization systems which make possible the proletarianization of all those tasks conducted at the highest levels of nervous system activity.” Crucially, it is the real-time abstraction of labor power as a potentiality, which we might think of as an unending stream of on-demand labor power, akin to electricity or water supply. This is mediated via chatbots and apps to create a highly alienated form of labor power that is, in a sense, the realization of cybernetic systems that attempt to closely couple machines and humans in tightly linked feedback loops. This describes the threat that Weizenbaum warned about in *Computer Power and Human Reason*.

The paradigmatic example is OpenAI's ChatGPT, which the company's CEO claims could transform capitalism: "[T]his is going to be a continual exponential path of improvement of the technology and the positive impact it has on society" (Altman, as cited in Konrad & Cai, 2023). ChatGPT is a chatbot built on a Generative Pre-trained Transformer, a type of large language model (LLM). The LLM was trained on real conversations, texts, and articles written on platforms such as Reddit. It has amazed contemporary observers with the seeming mechanization of the production of written text by prompting the system with input and questions. Meanwhile, the power of the system is actually hidden behind the chatbot's "interface." This notion of not only aggregating human cultural production through software but also treating it as a standing reserve for a computational system is indicative of the kind of cybernetic thinking prevalent in a computational society that Weizenbaum presciently warned about. In many ways, this makes human activity discrete. However, it is also the dehumanization of humans via a computation layer used to mediate cultural production more generally. It also shows how the interface acts to reify the social labor undertaken beneath the surface, such that the machinery may comprise literally millions of traces of humans' writings and conversations, all without them being aware of this or consenting to it.

This opens up the possibility of automated cultural production that is managed, controlled, monitored, and disaggregated and re-aggregated on demand. When operationalized by capitalist corporations, it is in danger of creating a social shock – new technical practices often fly under the radar of labor laws and protections – while also being hugely profitable. This can lead to distorting effects on the wider economy, a function of the lack of regulatory control by governments or oversight by labor organizations, such as unions. It is likely that we will see future social conflicts associated with the expanded use of automation systems that function in this way. In effect, they create a sense of algorithmic fetishism, with labor hidden behind algorithms, meaning that the programmer or user who interacts via dashboard interfaces and programming code need not acknowledge its social character. For the worker on the other side of the interface, the demands on agency, physical labor, emotional control, and self-discipline are likely to create severe psychic tensions (in terms of shifts between anomie and fatalism). A single bad review or rating from a customer or client can instantly cause termination of employment, with reasons seldom given. Whether this will create the conditions for a politics of necessity and a sociological reflexivity remains to be seen, but this certainly represents a potent future source of labor disenchantment beyond that of the traditional working class. We see a similar example of this in the current fashion for chatbots based on text interfaces, not unlike ELIZA, that partially automate and partially use humans to manage customer queries and problems (Marino, 2006). Needless to say, the idea that ELIZA would inspire computational systems that create cognitive factories would have horrified Weizenbaum.

Using this new paradigm, systems that use generative AI suggest the possibility of a new class of knowledge-powered interfaces. Here, the machinery is obfuscated by the interface and may be human labor, a chatbot, some other form of AI. From the perspective of the interface's user, there is no difference. This procedure reifies the relationship and creates a command–execute relationship between the user and the underlying process. The obfuscation of the means (regardless of the exploitative relationship it may be encoding) and the social and political consequences of the rationalization and abstraction of real human beings within computational systems echoes Weizenbaum's 1976 observation:

[A]n individual is dehumanized whenever he is treated as less than a whole person. The various forms of human and social engineering... do just that, in that they circumvent human contexts, especially those that gave real meaning to human language. (p. 266)

5 Reflections on the Implications of Generative AI

By examining the antecedent arguments and discussion around early computation, we are given a useful way of reflecting on current predicaments and problems that emerge in our rush to rationalize and computerize society. As a computer scientist, Weizenbaum was willing to think reflectively about his own practice and the implications of his work:

[W]hat should this teach us, particularly with respect to the question of at least preserving if not enhancing human choice in human affairs? Certainly that the construction of reliable computer software awaits, not so much results of research in computer science, but rather a deeper theoretical understanding of the human condition. (Weizenbaum, as cited in Rosenberg, 1980, p. 103)

Although ELIZA might appear primitive today, it encapsulates many of the design decisions that continue to be manifest in the systems we use, namely, decisions concerning how humans and machines interact and the extent to which computation should be allowed to govern our lives and minds. It also demonstrates that early software genealogy is crucial for understanding the extent to which our assumptions about the past and our ideas about how technology can and should develop must always be challenged. Although we might not think of ELIZA as AI today, even if that mistaken assumption was made in the past, we should be aware of the similar instrumental rationalities embedded in machine learning and today's AI systems.

Today's computer science milieu differs substantially from the one that Weizenbaum inhabited. He worked in an academic environment, albeit funded as part of the Project MAC program by the US Department of Defense, exploring technologies (e.g., ELIZA) through the lens of experimental research. Today, AI is increasingly subsumed by the needs of capitalism (Berry, 2021; McQuillan, 2023). Most AI development tends toward the historical norm of computation, including the capacity to separate control from execution at a technical level. A critical history of computation shows that this model is often applied to processes that align with capitalism, with an implicit set of assumptions built into the designs, such as an a priori assumption of the superiority of markets for structuring social relations. Second – and to my mind, not unrelated to this – current approaches to understanding AI tend to encourage metaphysical or formalist justifications and explanations of its functioning. This can lead to the valorization of the mathematization of thought, which sees the formalization of knowledge as not just one approach to thinking about AI but the exemplary approach. This can engender a theory of computation that leans toward idealism rather than a focus on who owns and controls the new “means of cognition,” echoing Blaise Pascal's 17th-century observations:

[There is a] difference between the mathematical mind (*esprit de géométrie*) and the intuitive mind (*esprit de finesse*): The reason that mathematicians are not intuitive is that they cannot see what is in front of them and, being used to the clear-cut, obvious principles of mathematics, and to draw no conclusions until they have properly understood and handled their principles, they become lost in matters which require intuition, where principles cannot be handled like that[...] These things are so delicate and numerous that it requires a very delicate and precise cast of mind to feel them, and to judge accurately and correctly from this perception. Most frequently it is not possible to demonstrate it logically, as in mathematics, because we are not aware of the principles in that way, and it would be an endless task to set about it. The truth must be seen straightaway, at a glance, and not through a process of reasoning, at least up to a point. So it is rare for mathematicians to be intuitive, and for the intuitive to be mathematicians, since mathematicians want to deal with intuitive things mathematically, and are ridiculous for wanting to begin with definitions, followed by principles[...] It is not that the mind does not do it, but that it does it silently, naturally, and simply. (Pascal, 1995, p. 151, translation adapted)

Without denying the immaturity of methods for the humanistic or social scientific study of AI or machine learning, I think that Weizenbaum would agree that it is precisely the assumption that mathematization or formalization is a precondition for understanding that is a grave error in thinking about AI. After all, Weizenbaum (1976) himself explained,

[T]hey may have begun by believing that the calculi they adopted were merely a convenient shorthand for describing the phenomena with which they deal. But, as they construct ever larger conceptual frameworks out of elementary components originally borrowed from foreign contexts, and as they give these frameworks names and manipulate them as elements of still more elaborate systems of thought, these frameworks cease to serve as mere modes of description and become, like Maslow's hammer, determinants of their view of the world. (p. 102)

With the advances in machine learning represented by systems such as ChatGPT, the danger is that the destruction of the capacity of human thought might be misunderstood as merely an accidental side-effect of AI. However, as with any technology, human beings have a choice about which technologies are created, how they are implemented, and the limitations and regulations around their use in society. Weizenbaum was prescient in understanding the challenge that ignorant, meaningless machines might have for our capacity for not only understanding and intersubjective communication but also our basic humanity. He identified the natural empathy humans have toward one another and the way that this can be misdirected toward computers and recognized the great temptation for technologists to use this empathy to mislead or deceive humans to make a profit, to nudge or persuade them, or to undermine democracy itself. We live in the future that ELIZA hinted at, and that Weizenbaum tried to warn us about. It is now our duty to heed his warnings and decide upon the limits of the computable.²²

²² Here, I understand "the computable" in the critical sense of understanding what kinds of problems are *appropriate* for computation, rather than the more technical definition that refers to what can or cannot be represented by an algorithm. In this context, the uncomputable would be those problems that society defines as inappropriate for computation.

References

- Anderson, C. (2008). The end of theory: The data deluge makes the scientific method obsolete. *Wired*. <https://www.wired.com/2008/06/pb-theory/>
- Arendt, H. (1972). *Crises of the republic*. Harcourt Brace Jovanovich Publishing.
- Berry, D. M. (2008). *Copy, rip, burn: The politics of copyleft and open source*. Pluto Press.
- Berry, D. M. (2011). *The philosophy of software: Code and mediation in the digital age*. Palgrave.
- Berry, D. M. (2014). *Critical theory and the digital*. Bloomsbury.
- Berry, D. M. (2021). Explanatory publics: Explainability and democratic thought. In B. Balaskas, B. and C. Rito (Eds.) *Fabricating publics: The dissemination of culture in the post-truth era* (pp. 211–233). Open Humanities Press.
- Boden, M. (1977). *Artificial intelligence and natural man*. MIT Press.
- Cayley, J. (2015). The listeners. <http://programmatology.shadoof.net/?thelisteners>
- Cerf, V. (1973). RFC 439 - PARRY encounters the DOCTOR. Faqs.org. <http://www.faqs.org/rfcs/rfc439.html>
- Crevier, D. (1993). *AI: The tumultuous history of the search for artificial intelligence*. Basic Books.
- Dembart, L. (1977, May 8). Experts argue whether computers could reason, and if they should. *New York Times*. <https://www.nytimes.com/1977/05/08/archives/experts-argue-whether-computers-could-reason-and-if-they-should.html>
- Fisher, A. W., & McKenney, J. L. (1993). The development of the ERMA banking system: Lessons from history. *IEEE Annals of the History of Computing*, 15(1).
- Hofstadter, D. (1995). *Fluid concepts and creative analogies*. Perseus Books.
- Irani, Lilly C. and Silberman, M. Six (2013) Turkopticon: interrupting worker invisibility in amazon mechanical turk. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). *Association for Computing Machinery*, New York, NY, USA, 611–620. <https://doi.org/10.1145/2470654.2470742>
- Irani, L. (2015). The cultural work of microwork. *New Media & Society*, 17(5), 720–739.

- Konrad, A., & Cai, K. (2023). Exclusive interview: OpenAI's Sam Altman talks ChatGPT And how artificial general intelligence can 'break capitalism.' *Forbes*. <https://www.forbes.com/sites/alexkonrad/2023/02/03/exclusive-openai-sam-altman-chatgpt-agi-google-search/>
- Marino, M. (2006). I, chatbot: The gender and race performativity of conversational agents [Unpublished Ph.D. thesis]. University of California Riverside.
- Maudsley, D., Greenberg, S., & Mander, R. (1993). Prototyping an intelligent agent through Wizard of Oz. In *Proceedings of the INTERACT '93 and CHI '93 conference on human factors in computing systems* (pp. 277–284). ACM Digital Library.
- McQuillan, D. (2023, June 6). Predicted benefits, proven harms: How AI's algorithmic violence emerged from our own social matrix. *The Sociological Review Magazine*. <https://doi.org/10.51428/tsr.ekpj9730>
- MIT (2008, March 10). Joseph Weizenbaum, professor emeritus of computer science, 85. *MIT News*. <https://news.mit.edu/2008/obit-weizenbaum-0310>
- Montfort, N. (2004). Continuous paper: The early materiality and workings of electronic literature. http://nickm.com/writing/essays/continuous_paper_mla.html
- Natale, Simone, (2021). The ELIZA effect: Joseph Weizenbaum and the emergence of chatbots. In S. Natale (Ed.), *Deceitful media: artificial intelligence and social life after the Turing test* (pp. 50–67). Oxford University Press. <https://doi.org/10.1093/oso/9780190080365.003.0004>
- Pascal, Blaise (1995). *Pensées and Other Writings*, Oxford University Press
- Pentland, A. (2015). *Social physics: How social networks can make us smarter*. Penguin.
- Proctor, P. (2015). What are the best Siri replies? *Quora*, <https://www.quora.com/log/revision/82373474>
- Retelny, D., Robaszekiewicz, S., To, A., Lasecki, W. Patel, J., Rahmati, N., Doshi, T., Valentine, M., & Bernstein, M. S. (2014, October 5–8). *Expert crowdsourcing with flash teams* [Paper presentation]. UIST: ACM Symposium on User Interface Software and Technology, Honolulu, HI, USA. <https://hci.stanford.edu/publications/paper.php?id=284>
- Rheingold, H. (1985). *Tools for thought: The history and future of mind-expanding technology*. MIT Press.
- Rosenberg, R. L. (1980). *Incomprehensible computer systems: Knowledge without wisdom* [Unpublished Masters Thesis]. MIT Laboratory for Computer Science.

- Sack, H. (2018). Joseph Weizenbaum and his famous Eliza. *Science Hi Blog*. <http://scihi.org/joseph-weizenbaum-eliza/>
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development*, 3(3).
- SRI International (2016). *About Us*. <https://www.sri.com/about>
- Stiegler, B. (2010). *For a new critique of political economy*. Polity.
- Tannenbaum, S. (2015). With one of its easter eggs, SIRI evokes the Firesign theatre, Hacker culture, a 1960s Chatbot, and Steve Jobs. *Medium*. <https://medium.com/@stannenb/with-one-of-its-easter-eggs-siri-evokes-the-firesign-theater-and-steve-jobs-86ea5b4874d3#.rn9smlfgm>
- Weil, P. (2015). Seriously writing SIRI. *Hyperrhiz: New Media Cultures*, 11. [doi:10.20415/hyp/011.e05](https://doi.org/10.20415/hyp/011.e05)
- Weizenbaum, J. (1962). How to make a computer appear intelligent, *Datamation*, 8(2).
- Weizenbaum, J. (1963). Symmetric list processor. *Communications of the ACM*, 6(9).
- Weizenbaum, J. (1966). ELIZA: A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 36–45.
- Weizenbaum, J. (1967). Contextual understanding by computers. *Communications of the ACM*, 10(8). <http://www.cse.buffalo.edu/~rapaport/572/S02/weizenbaum.eliza.1967.pdf>
- Weizenbaum, J. (1972). On the impact of the computer on society. *Science*, 176.
- Weizenbaum, J. (1976). *Computer power and human reason: From judgement to calculation*. W. H. Freeman and Company.
- Weizenbaum, J. (1987). Not without us. *ETC: A review of general semantics*, 44(1), 42–48.

Date received: July 2023

Date accepted: July 2023