The Role of Computer Science in the Age of Artificial Intelligence

Hannes Werthner[1]
TU Wien and Institute for Human Sciences (IWM)
hannes.werthner@tuwien.ac.at
Version Jan 17, 2026

Abstract

Developments in the field of sub-symbolic Artificial Intelligence have surprised and continue to surprise us with continuously new results. They go far beyond text generation and automatic real-time translation. The potential is enormous, but there are also massive critical issues. With this and as never before, computer science is now at the center of public discussion, right up to the political level of geopolitics. At the same time, a methodological shift in computer science is taking place: The focus is shifting from correctness and certainty to uncertain statements based on probabilities. What does this mean for computer science? The discipline is challenged methodologically from "inside" and at the same time from "outside" by its growing importance implications. How to deal with this situation? Does digital humanism provide a framework?

Introduction

The question regarding the new role of computer science (CS) is a difficult one; one is tempted to ask: is any role left[2]. Is CS as a discipline in crises, as noted by (Vardi, 2024), challenged by "internal" methodological developments and external ones by its own success. What is the role we play, which one should we play and what is our future?

Developments in the field of sub-symbolic Artificial Intelligence (AI), which are no longer so new, continue to surprise us with continuously new results. It is no longer just about text generation and automatic real-time translation (which was very difficult or even impossible in the past). The potential is enormous, but it also raises massive social, economic, political or environmental issues – up to the question of "What is humanity and its role?" As never before, CS and its impact is now at the center of public discussion – right up to the political level of geopolitics and global regulation.

Traditionally, CS has focused on a rather pragmatic orientation toward problem solving. At the theoretical level, it was based on formal models that emphasize correctness, certainty, and provability. These foundations are increasingly being replaced, or at least complemented, by methodologies that yield probabilistic and inherently uncertain results. At the same time CS sees the need to interact or even integrate methods from the social and human sciences.

---

[1] The paper is based on my keynote "The Role of Computer Science in the Age of Artificial Intelligence" at the dighum research conference Nov 2025, Vienna – https://link.springer.com/book/10.1007/978-3-032-11108-1. The writing was an AI experiment: the lecture's audio tracks were automatically transcribed, then simultaneously edited/texted by ChatGPT and Gemini, and then revised by me. Ultimately, the entire text is mine again. So, it doesn't quite work yet.

[2] Virginia Dignum (University of Umeå, Sweden) in a private communication.

Thus, CS faces methodological challenges emerging both from within (regarding its own epistemic assumptions) and from outside (through its societal impact). How should the discipline navigate this evolving landscape? And, in the context of this publication, but also in general as digital humanism is becoming a global guiding principle in the digital world, does digital humanism offer an adequate framework for the future of our discipline.

In the following, I begin by shortly highlighting major achievements in both Ai and CS, followed by a discussion, or better reflection, of both fields, their methods, and paradigms. This allows for a subjective conclusion about how the two fit together and complement each other. However, the nature of CS will change considerably. I then identify key challenges for academic institutions and conclude by considering the extent to which digital humanism, as an intellectual and normative approach, can offer productive guidance for the future of CS.

The existing literature on these questions is surprisingly very limited – only a few studies exist that deal with education and curriculum development. Much of my discussion is therefore based on conversations with colleagues, notably early-career researchers.[3] My special thanks go to them.

Success Story

In the following, I provide a brief overview of successful developments in both fields, AI and CS.

Artificial Intelligence

The current development of AI shows a series of unexpected and significant successes, surprising even renowned key researchers who laid the foundations of these successes[4]. Although our discussion focuses primarily on sub-symbolic AI (e.g., deep learning models), the big progress in various fields requires a broader, more comprehensive view. Today's AI systems have shown a performance that significantly surpasses previous expectations and, in many instances, exceed human-level proficiency across a range of cognitive tasks, for example:

- Advanced Problem-Solving: AI is now able to effectively solve mathematical problems at the doctoral level and actively targeting hard problems such as programming SAT (Satisfiability) solvers. These systems have shown results similar to or even better than human written programs. And AI systems are able to solve complex logical puzzles.
- Superiority in University admission tests: AI models outperform humans in various cognitive activities, including standardized admission tests and challenging content generation tasks.

[3] G. Heiler, B. Krüpl-Sypien, J. Oster
[4] Geoffrey Hinton has mentioned being surprised multiple times, e.g., in CBS Saturday Morning. Season 12. With Jacobson, D. (host); Silva-Braga, B. (reporter); Frost, N. 25 March 2023.

- Multimodal capabilities: The scope of AI application has expanded beyond capabilities like text generation and real-time machine translation to include high quality image understanding and generation.
- Labor productivity: Noy and Zhang (2023) documented that workers with initially lower productivity demonstrated improved performance on writing tasks when granted access to large language models like ChatGPT. And Brynjolfsson et al. (2025) found an improvement in working labor productivity when ChatGPT was used in a complementary role.
- Scientific research: AlphaFold (from Google DeepMind) achieved highly accurate prediction of protein foldings, which may indicate at a paradigm shift in biology, and science in general,
- Prediction: AI-based systems, such as those used for hurricane forecasting in Jamaica, have demonstrated superior predictive accuracy compared to established classical meteorological models.
- Games: AI demonstrates clear superiority over humans in complex strategy games such as chess and Go.
- Complex task execution (Agentification): The development of so called "Agentification" allows AI systems to decompose and autonomously manage complex tasks involving multiple dependent subtasks.
- Scientific awardss: These achievements were recognized by several scientific awards of highest level: The 2018 Turing Award was given to Geoffrey Hinton, Yoshua Bengio, and Yann LeCun for their work on deep neural networks and deep learning, and the 2024 Turing Award to Andrew Barto and Richard S. Sutton for their fundamental work on reinforcement learning
  The 2024 Nobel Prize in Physics was awarded to John J. Hopfield and Geoffrey Hinton "for fundamental discoveries and inventions that enable machine learning with artificial neural networks"; and in Chemistry to Demis Hassabis and John M. Jumper (together with David Baker) for "protein structure prediction," a breakthrough made possible by AI / machine learning methods. The Nobel Prize Committee has thus officially recognized AI innovations as worthy of the "Nobel Prize" – similar to "hard" sciences such as physics or chemistry.

In general, one AI can be regarded as a general purpose technology (like IT) with almost unlimited areas of application. I do not discuss any of the many critical issues such as bias, or misinformation here (see later chapter on AI), but only point at one serious problem in scientific publications, where AI conferences are inundated with peer reviews written by AI.[5]

[5] Nature News 1.12.2025. Major AI conference flooded with peer reviews written fully by AI. www.nature.com/articles/d41586-025-03506-6

Impact of Computer Science

We live in a physical – virtual world: both “worlds” are increasingly intertwined. The Internet and the web function as the operating system of our society, of the world. What we call digital transformation, constitutes a multifaceted techno-social and socio-economic process, which began over 70 years ago and is now coming to the surface, with AI as its last appearance.

A simple everyday observation illustrates the depth of this transformation: in metropolitan public transport systems, most passengers today focus on their mobile devices, interacting predominantly with the digital rather than with other humans, i.e., passengers. Only fifteen years ago, the same spaces were dominated by printed newspapers. This comparison shows both the rapidity of technological change and the profound disruption experienced by traditional industries, such as the newspaper industry.

Historically speaking, it was not a long journey from the ENIAC (1946) as the first electronic calculator (computer) to the first University degree programs in CS in Austria in the 1970s, where today CS is even referred to as the Latin of the present.[6] And today it has even become an important topic in politics and geopolitics.

Phenomena such as the World Wide Web, cyber-physical systems, and the Internet of Things show us how far CS has developed in its short history. It has been a journey from stand-alone computers to the global operating system of our society, leading us into the midst of another industrial revolution, with the digitization of content and the automation of work and thinking. This global operating system connects and permeates everything: work, leisure, politics, personal, professional, and private life. At the same time, these devices are becoming invisible and increasingly “disappearing”: this is therefore an almost dialectical process of encompassing everything and disappearing at the same time. We have witnessed a metamorphosis from a computer to a pervasive global media and communication machine.

Economically, it went from *IT supports the business* to *IT is the business*, with a highly concentrated platform economy. The significance of this is also reflected in the ranking of companies with the highest market capitalization: the majority of the top companies are platform companies such as Apple, Google, Amazon, Facebook, and Microsoft. The value of these companies is based primarily on information (mostly not even their own, but information about and from their users) and their user network. They process this data, draw conclusions from it, and sell this “refined” information. The value and importance of these companies stems from the processing of information, not from “real” products and goods.

Two Sides

AI and IT in general have achieved remarkable successes and continue to show considerable potential. IT not only supports the functioning of modern socioeconomic systems, but also expands their capacities in unprecedented ways. The dissemination of rich, freely accessible information has become an indispensable public good, enabling new forms of knowledge

[6] Antonio Loprieno, Egyptologist and former chair of the Austrian scientific council.

sharing and collective action. Furthermore, the internet and social media play a crucial role in supporting citizen participation and organizing civil society. Examples range from the "Sardines" movement in Italy and the protests in Hong Kong to social and political campaigns in South Africa and Zimbabwe to coordination efforts among Amazon workers. Digital infrastructures have become indispensable for addressing fundamental societal economic and environmental challenges (see the UN Sustainable Development Goals) and enabling large-scale collective responses.

At the same time, however, there are enormous negative effects and major risks associated with this development, as T. Berners-Lee has already pointed out with his *The system is failing*.[7] The following is only a partial list; a discussion of these points would go beyond the scope of this paper (see Werthner 2025):

- Concentration and monopolies in the Web
- Sovereignty of states as well as individuals
- Fake news – with its impact on political discourse
- AI and decision making (the future role of humans)
- Privacy issues and surveillance
- Robots and warfare / weaponization of AI
- Work and automation
- Sustainability and environment

These two sides can also be viewed as the dialectic or ambivalence of digitalization; a positive thing can turn into its opposite; but that is not the end of the negative. Changing this is the task of digital humanism. A final note to this ambivalence of IT: History teaches us that social benefits are not automatically guaranteed. They require organizational, social, and political actions and measures (Acemoglu and Johnson, 2024).

The Computer

The following is a brief reflection on the "mechanical" basis of this development, the computer. It can be described as a general-purpose automaton that, as the only automaton, can control itself via software and can thus be "instantiated" via software to become a specific problem-solving machine: it can become a control device for a power plant and at the same time a social media tool or a large language model. This general-purpose machine has the unique ability to independently change and control its own behavior based on external inputs and its internal states; in other words, it can act quasi in a self-reflective manner and thus simulate intelligent behavior.

[7] Tim Berners-Lee: The system is failing. The Guardian, 16.11.2017. www.theguardian.com/technology/2017/nov/15/tim-berners-lee-world-wide-web-net-neutrality

The basis, also of CS, is a computational model, i.e., the mapping of solution steps onto an (abstract) machine and conversion into a corresponding abstract or concrete algorithm in the form of written code, an algorithm being a finite, deterministic step-by-step set of instructions to solve a specific well-defined problem.

Surprisingly, the computer as such is a relatively simple machine, but being able to solve highly complex problems. Let us take a look at the two most famous models of a computer, both being at the "heart" of CS.

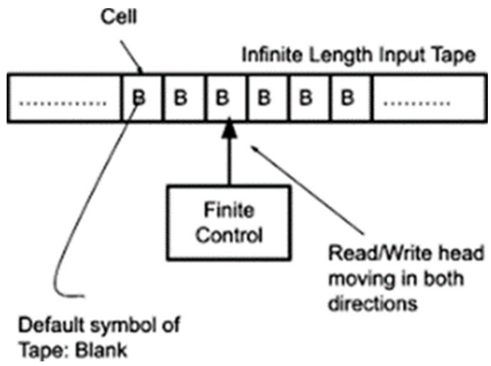

Figure 1: Turing machine

The Turing machine (Turing, 1936) is an abstract machine, a mathematical model of computation, it provides a formal definition of the concept of an algorithm and also computability, i.e., what can be computed or solved. It has a simple mechanism: an infinitely long tape (serving as input, output and memory) divided into cells, a read/write head that can read and write symbols on the tape one cell at a time, and a finite control unit (the program) that also stores the machine's current state. The program is simple, it is a finite table of rules, which define the machine's next action such as write a symbol, move the head left or right, and transition to a new state. These actions only depend on its current state and the symbol it is currently reading. Despite this simplicity, the so-called Church-Turing thesis states that any computation that can be performed by any computer, can also be performed by tis simple Turing machine. This also defines the model of all general-purpose computation, as well as its the fundamental limit. What cannot be solved by a Turing machine, no computer can do.[8]

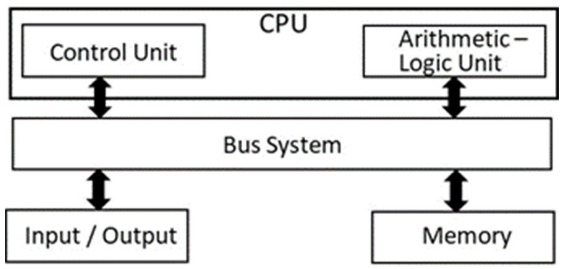

Figure 2: John-von-Neumann computer architecture (von Neumann, 1945)

[8] This was demonstrated using the "haltingproblem": is it possible to determine for any given program whether it will eventually finish its calculations or run in an endless loop. With his machine, Turing showed that there is no general algorithm that can solve the haltingproblem for all possible programs.

The second example of simplicity is the von-Neumann computer architecture of a stored-program, i.e., programmable, computer, this architecture is still found in modern computers. It consists of a memory, a control unit, an arithmetic–logic unit (ALU), input and output. Interestingly, as early as 1834, the Englishman Charles Babbage designed a similar model of a computer with his Analytical Engine (which was never built).[9] The computer architecture has a great similarity with the structure of a company: The control unit orchestrates the work and interprets the computer program step by step, it corresponds to the management of a company. The ALU is doing the work, it executes the individual program commands, like workers in a company. The memory contains the data to be processed – corresponding to a company's warehouse. The input of data and the output of results correspond to the purchasing and sales departments. In Figure 2, the ALU and control unit are combined into the so-called Central Processing Unit (CPU). The bus system is used for internal data exchange, it corresponds to the internal communication of a company.

The trick with computers is that both programs and data are stored together in memory and are only "treated" differently when they are executed. This makes the computer a universal machine.

Computer Science

CS can be regarded as the science of today's information society. Its methods and paradigms influence how we perceive the world and how we think about it, its artifacts are changing the world.

Surprisingly, it is not so clear what CS is, there are many, nearly countless, definitions. The first who mentioned the term Computer Science was George Forsythe as early as 1961 (Forsythe, 1961): " ... In spite of the diversity of the applications, the methods of attacking the difficult problems with computers show a great unity, and the name of Computer Sciences is being attached to the discipline as it emerges." Donald Knuth referred in his Turing lecture 1974 to it as the "study of algorithm", closely related to mathematics (Knuth, 1974). It should also be noted that Knuth also speaks about it is an art form, in an aesthetic sense, similar to the work of architects. Good programs show a beauty, which can be compared to fine art and beyond pure functionality.[10]

The following definitions, all related to the ACM[11] environment, show the development from a computer, i.e., artefact, focused approach to a broader perspective:

[9] It is unclear whether von Neumann knew of Babbage's work, but this is suspected.

[10] Donald Knuth also wrote "bible" of computer science, a fundamental, multi-volume encyclopedia that provides a rigorous and comprehensive treatment of data structures and algorithms, and called it "The Art of Computer Programming." (Knuth, 2023)

[11] The ACM (Association for Computing Machinery) is the world's largest educational and scientific computing society, dedicated to advancing computing as both a science and a profession. Its members are institutions, companies and individuals.

- Norman E. Gibbs and Allen B. Tucker, 1986: Computer science is the study of algorithms and data structures, including their formal properties, mechanical and linguistic realizations, and their applications.
- Peter J. Denning, et al. 1989: Computer science is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application. Here it extends to processes and information transformation
- Allan Tucker, et al. 2006: Computer science is the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society. Here the definition also refers to human aspects.

This list shows very well that definitions evolve, become broader, from focusing on algorithm to the process of information transformation and incorporating human aspects. This development is also well shown by Kristen Nygaard's[12] (Nygaard, 1986) comprehensive and far-reaching definition, which, interestingly, dates back to the 1980s: "Informatics is the science that has as its domain information processes and related phenomena in artifacts, society, and nature." According to him, it deals with the understanding fundamental principles of information processes, both natural and artificial. Thus, CS is therefore not only concerned with a specific machine (the computer). This is also in line with statements such as that made by Nobel Prize winner David Baltimore: "Biology today is an information science."

The term "Informatics" is broader than "Computer Science" and mainly used in Europe; is also more interdisciplinary; see also the so-called "Bindestrichinformatik"[13] (e.g., business informatics), which is common in the European University programs. In the following, I do not distinguish between the two terms.

There are three so-called founding disciplines of CS, with their different conceptual approaches and methodologies – resulting into a rich overall methodological framework:

- **Mathematics and** Logics (formal basis): Computability, algorithms, complexity
- **Engineering**: from formal statement and requirements to design and implementation, system construction, software engineering, architecture, including also the design science approach of (Hevner et al., 2004).
- **Empirical approach**: routed in natural science; from formulating hypotheses to testing, with experimentation, measurement, and observation

[12] He received, together with Ole-Johan Dahl, the Turing price for developing of SIMULA, the first object oriented programming language, primarily designed for simulation purposes.
[13] With this I refer to European study programs such as business informatics, media informatics, or medical informatics, all combinations of CS and a dedicated application field.

From a technical perspective, systems consist of a stack of diverse components, hardware and software, the former not only computers, but also any other machines. One may only think of transportation systems or individual vehicles. In these systems tasks are increasingly being delegated to the software, leading to an increasing virtualization. Today, a car, for example, can be seen as a mobile phone on wheels full of software rather than just a means of transportation. This leads to increasing virtualization. Thus: **Everything touched by software, becomes a computer and part of a mega machine.**

From a methodological point of view CS "shows" two faces;

- *CS as subject*, with research and development in areas such as algorithm, modelling and design, information representation, programming languages, complexity theory, software engineering, …

- *CS in subject*, as a tool, a methodology and even an ontology in other sciences as well as application fields

CS has provided a new perspective on natural and man-made phenomena, thereby enabling other disciplines to further develop. Scientific research now has access to an "info-computational" theory of science with a new ontology (how we see the world), epistemology, and methods. This also shows that CS is inherently interdisciplinary and has also become a foundational discipline. And CS creates new things, both virtual and real, with almost no physical limitations. In this respect, it bears similarities to art: "Everything is possible."

There are two further facets to CS: solving specific problems and, at the same time, attempting to understand and automate human thinking. Let's take von Neumann again, who built concrete computers for problem solving (e.g., weather forecasting and simulation) and at the same time developed the theory of self-reproducing automata (von Neumann, 1966). Or Alan Turing with his concrete computer Colossus, which he used to crack German military codes during World War II, and his Turing test, with which he defined formal criteria for evaluating the intelligence of machines.

This shows that it is difficult to separate basic research from applied research, aimed at solving specific problems. But one should note that basic research plays an important role in this mix. This is illustrated by the so-called "tire track model," which empirically proves that it takes an average of 20 years for a basic research result to turn into a large billion-dollar market (e.g.., mobile devices). This is also evident in the current AI developments, whose roots go back decades. The example of machine learning and data science illustrates very well the aforementioned characteristics: i) the combination of an empirical scientific approach (learning from data) with formal methods and algorithms as well as programming and system development; ii) the intertwining of application and research, with research in this area relying on real application data and its understanding.

I would like to conclude this paragraph with one final observation: CS is also a discipline of efficiency, i.e., solving problems quickly with as little effort as possible, and of effectiveness, i.e., identifying whether a problem can be solved . Take complexity analysis, for example: examining how “expensive” a solution is, or computability theory: which problems can be solved and where are the theoretical limits of computation? There is little work on the subject of resilience, as in our economy[14]. With this orientation, CS reveals a close relationship to neoliberalism and current developments in our society.

Artificial Intelligence

AI, as a field, is inherently undefined, highlighting the absence of a universally accepted definition and the resulting ambiguity in what should or should not be classified as AI (Haigh, 2023, 2025). The term “Artificial Intelligence” often functions as a 'brand' encompassing vaguely related technologies, primarily driven by market hype. This lack of consensus not only opens the door for diverse perspectives, but also broadens the scope of topics included within its domain. For example, classifying recommender systems as part of AI illustrates this ambiguity, as such systems encompass multiple disciplines, including CS, cognitive science, and also marketing.

In everyday language, AI could be described as a simulation of human thinking; machines we have created exhibit "intelligent" behavior (measured against human intelligence). It's something like the "automation of thinking," materialized in computer code. But it leaves open the question of what intelligence actually is. Is it solely cognitive processes, all in the brain, but isn't it also embedded physically and socially?[15]

Essentially, or pragmatically speaking, it is indirectly defined by the application of two classes of methods: logic-based AI and sub-symbolic or data-driven AI. Since its beginnings in the 1950s, a dominance of the logical approach can be observed, while today, since around 2015, data-driven approaches prevail. One could even say that this is now considered THE AI. This methodological shift is also leading to a cultural change within the research community, what would be worthy of a sociological study.

Definitions of AI

Looking for definitions it makes sense to go back to the first workshop on AI in 1956, which created the term “Artificial Intelligence” and also defined it in its workshop proposal and request for funding (McCarthy, et al (1955): “….. conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it”.

Workshop participants, and later key figures in research, defined it as follows: Marvin Minsky (1968): “The science of making machines do things that would require intelligence if done by men.” And John McCarthy viewed AI as both a scientific discipline (understand intelligence) and an engineering field (build intelligent systems) (McCarthy, 2007). Wikipedia has the following definition[16]: “AI is the capability of computational systems to perform tasks typically associated

[14] With a famous exception, the Internet. Given its background in military research, it's not hard to guess why.
[15] A discussion of the concept of intelligence is not the subject of this article.
[16] https://en.wikipedia.org/wiki/Artificial_intelligence

with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making."

These various definitions are relatively vague and, more importantly, indirect: they refer to human intelligence, for which there is still no single, universally accepted definition, especially in all its facets. They all, however, relate to machines (computers) and their engineering. AI can thus be seen as part of CS.

From a more pragmatic perspective, AI is defined by the use of two method classes: the logic based and the learning-based approach. In the following we discuss only the latter one, it is by far the dominant one today.

Learning / sub symbolic AI[17]

In this subfield of AI methods moved from Machine Learning (ML, e.g., neural networks) over deep neural networks (with a high number of intermediate layers in the network, this improving the learning capacity of the approach) to generative AI, i.e., large language models (LLM) – it moved from classification of content to the generation of "new" one.

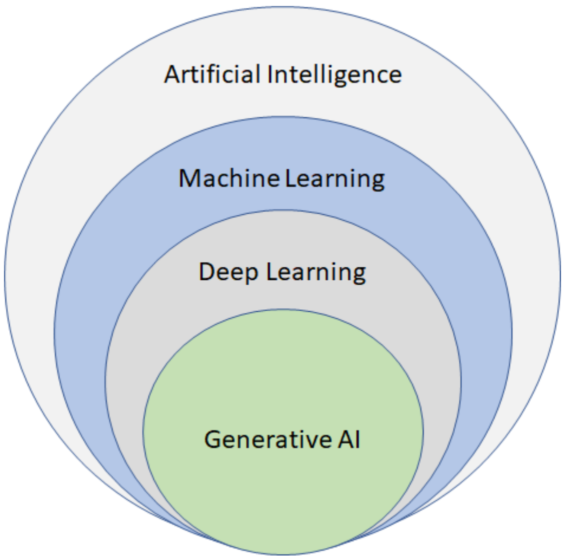

Figure 3: Classification of sub-symbolic AI

In ML, a system learns patterns from given data (with labeled or unlabeled examples) during a training phase in order to classify new data. An LLM learns statistical "representations" of our language and, in the case of multimodal data, also of the "richer world" described in the given data, to generate a variation or re-combination of existing content. The success of LLM, or the learning approach in general, is based on three factors: the availability of accessible data (the web and search engines), new hardware for parallel processing, such as graphics processing units (GPUs), and methodological advances (deep learning and LLMs). In some ways, it is more of an engineering than a theoretical success.

[17] A description of the different AI methods is beyond the scope of this paper.

An LLM "learns" language without knowing the grammar or its rules at all, i.e. how to form sentences, what an adjective or a verb is, etc. It is a statistical representation of our language based on the data available for training.

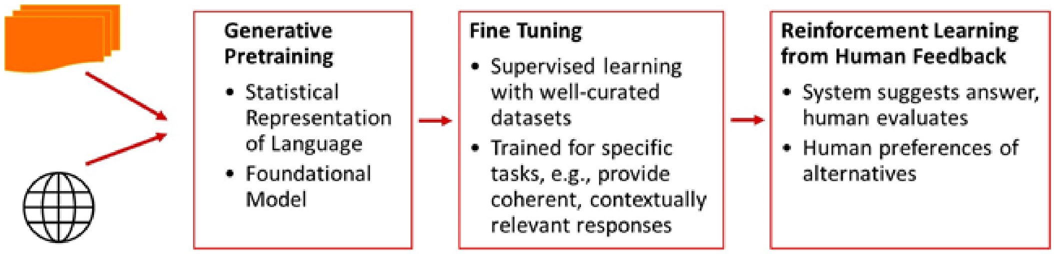

Figure 4: Three phases of "learning" of ChatGPT (Generative Pretrained Transformer): generative pretraining, finetuning, and reinforcement learning from human feedback (see also Werthner, 2025)

Figure 4 shows the learning sequence of ChatGPT, starting with the process of learning how the system can generate text itself, then it is fine tuned to a specific task and finally it receives feedback from people to get even better. The first step is generative pretraining: here the system recognizes words (i.e., tokens[18]) and connections between words based on a huge amount of text. Basic patterns and relationships in the language are recognized. The result is the so-called foundational model, which can then be adapted to a variety of tasks. This next step is fine tuning by supervised learning with well-curated datasets. In this phase, the model is trained on specific tasks, such as a conversational chat. The idea is to better tailor it to the user's expectations and reduce it to the defined task. In the third step, the system uses so-called reinforcement learning from human feedback (RLHF), in which human trainers evaluate the answers suggested by the system and thereby improve them. The human feedback is used to train a reward model that scores the outputs based on desirability.

In essence, a LLM is trained to predict the next word, and when a text fragment is entered, they output which word could follow next and with what probability. The “knowledge” of the system is not explicit (e.g., grammar tree or relational rules), it is the billions parameters linking tokens in the statistical network. There is no explicit model.

In some sense an LLM is the creative, but context- and “content-free” combination of mathematics, statistics and probability theory, and CS. This can also be seen by the list of methods and approaches (from these disciplines) applied in LLMs:

- Tokenization and word embedding
- Transformer – systems architecture
- Software engineering
- Gradient method / optimization technique

[18] A token is the smallest unit of data, into which text is broken down so that a machine learning model can process it.

- Data structures
- Distributed systems and high-performance computing
- Evaluation metrics

In table 1 I compare LLM with the classical algorithmic approach in CS. One can see clear pairs of opposites in all aspects, but at the same time these also expand the computational space.

Aspect	**Classical algorithmic approach**	**LLM**
Knowledge source	Explicitly programmed rules	Implicitly learned patterns
Knowledge Representation	Explicit symbols	Implicit embeddings and weights
Control	Top-down (humans)	Bottom-up (learned by the network)
Behavior	Deterministic	Probabilistic
Traceability	High	Low
Adaptability	Low (reprogramming necessary)	High (fine-tuning, prompting)
Scaling	Needs reprogramming	Automatic by new data
Validation	Formal proofs	Empirical performance
Computation	Deterministic execution of code	Probab. inference, guided by prompts
Interaction	Programmer defined	user guided / prompting
Goal	Deterministic behavior	Statistical performance (possible future)

Table 1: Comparing the algorithmic approach with LLM

Regarding human computer interaction, one could say that we move from an interaction defined by specialists, i.e., programmers, to one enabling a simple text interaction by everyone. And this new type also makes fun as the very high number of users show. The question is, if this is more democratic?

The nature and focus of the human-computer relationship shifts with the transition from deterministic to probabilistic systems. Previously, we delegated problems to machines to obtain "safe" and guaranteed solutions (with our truth machines). Now we have machines that invent stories, and we must verify the results. The new role of humans could be to act as prompt engineers, guiding the system in finding solutions, and as the verifier of the outcomes.

In the final part of this chapter on AI, I will briefly list some critical points without going into detail and then outline some very general future research and development issues. Critical points are, among others

- LLMs function as block boxes, one cannot, at least not easily, explain how or why they arrive at a specific answer

- Bias, as they produce unfair or prejudiced results, they mirror and amplify the historical stereotypes, social inequities, and data gaps found in their training data

- LLMs are story machines, no truth machines, they “hallucinate”
- Environmental issues such as energy consumption and water consumption and many others……..
- Problems in general reasoning with so called large reasoning models (LRMs) (Shojaee et al, 2025), where complex problems lead to a “complete accuracy collapse”. Such complex tasks are, for example, the River Crossing puzzle or the Tower of Hanoi. These systems seem to have problems with backtracking.[19]
- In addition, there are emerging critical features such as: LLMs seem to defend themselves if they are “threatened”; it is not easy to change their “values” and judgements, once trained; they tend to lie if they seem to need it.[20]

Building reliable AI will require something far beyond the current state of research, and integrating (among others)[21]

- World models that can predict and reason about real world situations
- Autonomous learning that discovers causal structures, that can be explained
- Systems that reason, plan, and act coherently within physical and ethical boundaries
- Evolutionary and meta-learning algorithms

In my view and on a bigger scale this calls for combining the logic and subsymbolic approaches, a vivid interface to other disciplines and will eventually lead to greater system architectures.

AI and CS

In the following, I will discuss that AI will not replace CS, but rather require its methods, along with some from other disciplines, while simultaneously expanding it and redefining its role. For this we start with a look at the programming approaches of CS, its founding disciplines and, finally, major principles of computing. Let us first consider the most important programming approaches and their application within AI:

---

[19] Backtracking: a problem-solving technique that explores possible solutions step by step and abandons a path as soon as it is determined to be invalid.

[20] The use of words like "lying" or "thinking" when describing the behavior of machines (only humans lie and think) shows that we still lack a proper vocabulary for this. We tend towards anthropomorphism.

[21] This is based on the newsletter of Ben Shneiderman and his summary of the lectures by Yann LeCun, Yoshua Bengio, Gary Marcus, and others at “Celebrating the 75th Anniversary of the Turing Test, organized by the Web science Institute, October 2, 2025:

- Functional Programming:[22] Many ML frameworks use functional constructs; this helps with encapsulation and parallelism

- Procedural: Procedural programs are used for orchestrating training pipelines, data loading, and deployment, they are the "glue code" of the systems

- Declarative: This approach is used to query data (e.g., in SQL), however, at the moment it seems to be less important, at least at the moment

- Object-Oriented: this paradigm is rather used on an implementational level, not so much on a conceptual one, where the world is seen as a set of interacting object

- A new approach is emerging, probabilistic programming languages (PPL), which rely and reuse the other classical programing paradigm.

As can be seen, all of the traditional approaches are reused. With regard to the founding disciplines, the "roots" of CS, I also see no replacement, but rather their continued use and expansion.

	Data-Driven AI	
Formal basis	This is the mathematical backbone with, e.g., information theory, probability, optimization, computational learning. It now formalizes what it means to "learn"	Deep learning architectures rely on linear algebra, optimization, and information theory
Engineering	This makes AI usable at scale – providing programming frameworks, distributed training, model deployment, and evaluation pipelines	LLM rely on massive infrastructure and distributed training, prompt engineering, model serving; also needed for scaling
Empirical	This is now central with validation via experimentation, cross-validation, etc rather than proofs. Empirical iteration drives progress	Quality arises from experimental iteration, dataset curation, and benchmark testing
Social Sciences / Humanities	*AI systems must align with human values, ethics, fairness, explainability, and societal impacts. Important human-in-the-loop learning and system use*	

Table 2: Founding Disciplines revisited

[22] I do not explain these programming paradigms, the reader is referred to CS text books.

There will be a significant expansion in the direction of social sciences regarding the effects on people and society, as well as in the direction of the humanities, for understanding and classifying results and in addressing ethical questions (see also Connolly, 2020).

A similar picture emerges when considering the “Great Principles of Computing” (Denning & Martell, 2015); these principles, quasi an ontology of computing, describe what machine-based computing fundamentally is.

Principle		Classical Examples	***Sub symbolic AI***
Computation	computational processes, what they can and cannot	searching for optimal solutions; traveling salesman,	neural computing
Communication	concern the transmission of data with reliable reception	transfer of information between entities, codes, protocols	NN: networks of com-municating units
Coordination	how autonomous entities work together	distribution, deadlocks, routing, parallelization	parallel and coordinated computation
Recollection	how computations store and recall information, effects on performance	virtual memory, caching, information structure, cloud services	“memory” is statistical and distributed
Automation	efficient computational ways to perform tasks (also automatically)	genetic algorithm, branch and bound, search algorithm	systems learn the automation rules
Evaluation	how systems perform, how much capacity needed to deliver results	evaluation and bottlenecks, performance metrics	systems measured behaviorally
Design	design of software/system which are dependable, reliable, usable, …	information hiding, modularity, software and system design	large-scale system architectures, e.g., transformer

Table 3: Principles of Computing

As you can see, these basic principles remain, they are, as is visible in the other cases, modified and also expanded.

As a Summary: AI redefines and extends CS

I conclude that data-driven AI and LLMs represent an evolution of CS, with its understanding of computation as learning. Scalability and performance are improved by exposure to further data, not by redesigning algorithms. And: it reinforces a data-empirical approach and the societal aspect of CS. Ai is the next step in the evolution of CS, with further to come. We see a transformation of CS from

The traditional role: "mother discipline"

AI as a subfield of CS, where it provided the essential foundations

over

The current role: "enabler" and "integrator"

AI is already providing a powerful toolbox to CS, and starts to automate CS itself: AI is beginning to automate the work of computer scientists. There will, for example, emerge new foundation of Software Engineering and AI modules will become new types of system component

to

The emerging role: "ethicist, architect, and innovator"[23]

AI and CS systems will have an increasing impact on our world, thus, the focus will increasingly shift towards requirements engineering and a broader stakeholder involvement, new architectures, rapid development processes, prompt engineering and more "user friendly" interaction. Coordination of fundamental building blocks, data governance, and data quality are crucial. We must strengthen interfaces to the other sciences as well as our communication with the public. This role goes far beyond technology.

Impact on Science[24]

The impact will be enormous, and cannot yet be fully assessed. In the following, I will discuss only a few aspects of this approaching future.

Nature of science

Science relies on models as fundamental tools for explaining phenomena and making accurate predictions, models serve as simplified and abstract representations of complex systems. Models provide a structured framework for understanding relationships between variables and generating forecasts. They are central, e.g., to hypothesis testing or experimental design. This is challenged by machine learning-based AI. It achieves impressive results by learning directly from data without the need for explicitly defined models and interpretable mechanisms. However, their lack of interpretability poses a dilemma. They do not satisfy the scientific goal of understanding *why* phenomena occur. We risk reducing science to a predictive exercise without deeper insights into causality. However, there are also opportunities for hybrid approaches, where data-driven techniques enhance or complement traditional models. We may shift to a

---

[23] One can also draw parallels here to the statements of Donald Knuth, with his view of CS as art.

[24] The following two paragraphs are based on the Digital Humanism workshop "A Paradigm Shift in Computer Science?", TU Wien, Vienna 2024. https://informatics.tuwien.ac.at/news/2800 and https://www.bing.com/videos/riverview/relatedvideo?&q=workshop+a+pardigm+shift+in+computer+science+tu+wien&qpvt=workshop+a+pardigm+shift+in+computer+science+tu+wien

more dynamic, data-driven discovery model, with AI playing a crucial role in every phase of the scientific workflow. Historically, research followed a structured path: observation, hypothesis formulation, experimentation, and analysis. Now increasingly AI technologies may be used to automate, enhance, and accelerate these processes. And AI can extract insights from complex datasets, revealing previously hidden phenomena. However, this shift raises concerns regarding reproducibility, interpretability, and scientific rigor, as decision-making increasingly relies on black box systems.

New generation of scientists

We will see a new generation of scientists, better acquainted with statistics, and open to a broader set of methodologies. Tomorrow's scientists will probably evaluate success based on performance metrics such as accuracy, precision, or recall, which fundamentally differ from understanding the underlying mechanisms and causal relationships. As a result, this new generation of scientists may exhibit different attitudes toward key aspects of research. For example, they may be more comfortable working with black-box models and accepting predictions without fully understanding the internal decision-making processes as long as the results meet performance criteria. This can, however, lead to a tension between achieving short-term practical success and pursuing long-term theoretical understanding, dominant in the current scientific paradigm.

The role of CS in science

The rise of AI reinforces the role of CS vis a vis other disciplines and society. Traditionally, CS primarily provided computational tools, algorithms, programming frameworks, and the computational paradigm that other sciences could adopt to solve their problems. However, AI is catalyzing the role of CS as the driver of scientific innovation, reshaping how knowledge is produced, issues are approached, and solutions are implemented across disciplines, see the recent breakthroughs in areas like biology, medicine, or environmental science. Moreover, AI's societal impact demands that scientists work closely with experts in other fields such as ethics, law, the social sciences, and public policy to address the broader implications of the digital. This points to the need for interdisciplinary collaboration. Thus, the role of CS extends beyond developing "good" technical innovations. It is part of a multidisciplinary endeavor that requires contributions from diverse fields to align technological progress with societal values and needs.

Impact on Academic Institutions

A reflection on the implications for and a discussion of possible responses by institutions should be based on several premises: a) Public universities are not profit-oriented organizations but must serve the common good. Education must not be reduced to a purely economic commodity. b) CS and AI go beyond the development of "good" technological innovations. They are part of a multidisciplinary endeavor to reconcile technological progress with societal values and needs. c) It should be accepted that private companies have an advantage w.r.t data, money, and infrastructure, but not when it comes to know-how. What does this imply for the different basis tasks of a University

- High-quality research is crucial, and the contribution of public, open research is essential. This, however, requires breaking down disciplinary silos. Besides CS's contribution to other disciplines, important topics often lie at the interfaces between disciplines.
- The teaching of the future will take place in a flexible, multidisciplinary space. The focus will be on fundamental concepts and methods (not on "programming" or other "skills"), with creativity and comprehensive systems knowledge being essential. The future curriculum will resemble a “Lego set” of various disciplines with flexibly combinable basic building blocks. However, one should be aware of the related risk of “knowing nothing about everything.”
- Universities play an increasingly important role in innovation and technology transfer. They contribute to the economic and social development of a society and are central components of location policy and the research ecosystem. These contributions are made at both the technological and educational levels.
- In a future shaped by information technology, we must reflect on our role in this process and act accordingly, with the motto: "We are not an NGO, but we care."
- Faster knowledge and innovation cycles present a further challenge as in the future, universities will need to develop lifelong and lifewide learning programs and provide an interface to their future students and schools. These will not be standardized programs, but rather ones tailored to specific needs and abilities.
- There is competition with private companies, not only for the best talent: This implies both competition and cooperation at the same time.

But the question may arise: Are the institutions aware of the extent of these developments and do they have the necessary strength to react accordingly?

Digital Humanism as Framework

In the following, I will briefly examine if Digital Humanism can serve as a guideline for the future role of CS (and not only for the digital in general), and how it can help in this regard. [25] Digital Humanism is defined as an approach to describing, analyzing, and, above all, influencing the complex interplay between IT and humans – for a society that fully respects universal human rights.[26]

It puts humans at the center of this "co-evolution" of humans and machines; humans are the architects of their lives. However, "human being" is defined more broadly here, we include our

[25] Within the context of this book on Digital Humanism, a brief description is sufficient.
[26] See the Vienna Manifesto on Digital Humanism. https://caiml.org/dighum/dighum-manifesto/

relationship to society and nature, as well as society and nature themselves. Humans exist within a social, political, and environmental context.

Digital Humanism is rooted in the Enlightenment; there is no being above humans – humans are the agents of their life; there is no technical or economic determinism. Today, in the digital world, it is not only important to eliminate negative developments, but also to shape technology and to act accordingly. Thus, Digital Humanism is not only a call to analyze and reflect, but also to take active action. It is a program with various dimensions, ranging from research and innovation up to political communication. And it underscores interdisciplinarity, the need for cooperation across disciplines, with CS playing a central role.

What does this mean for CS:

It is recognized as core discipline, probably even THE core discipline, it connects the various parts and areas of this development. However, we must be aware of that as this comes with a great responsibility. Although CS is of central importance, it is not enough on its own; it is only one part of an overall approach to solving critical problems and shaping the world. We need to understand other disciplines and learn from them. Ultimately, interdisciplinarity and the interfaces with other disciplines and their methods are crucial.

We as scientists, as individuals, as institution and as discipline are part of our society, which is fundamentally changed by our activities, thus, the role of CS extends beyond the development of "good" technologies, it's about social innovation. However, we mustn't forget that we have to develop good technologies. And digital humanism can teach us another lesson: future activities should cover various dimensions, from research and policy advice to interaction with the public.

Conclusions

We have to recognize that we, computer scientists, are pioneers, enablers, and drivers of digital transformation. And we should not forget that further CS technologies are on the horizon; on the contrary, we have not yet reached the end of the road, never will we.

As we have seen, AI does not replace CS; it redefines and expands it, the basic principles remain the same. With this we are taking on an additional, new role: computer scientists are becoming the architects, ethicists, and guardians of future intelligent (and non-intelligent) systems. But this is also a great responsibility. It underscores the importance of social issues and human needs that we must pay attention to when developing new systems, trying to integrate all stakeholders and to interact with other disciplines.

We face both an internal challenge regarding who we are now, and an external one due to our growing importance. This makes us vulnerable, but at the same offers great potential to strengthen CS. It is important to reflect on this, to be aware of it, and to act accordingly.

References

Acemoglu, D., Johnson, S. (2024) Power and Progress: Our Thousand-Year Struggle Over Technology and Prosperity. Basic Books, 2024.

Brynjolfsson, E., Li, D., Raymond, L. (2023). Generative AI at Work. The Quarterly Journal of Economics, 140 / 2, 2025.

Connolly, R. (2020). Why Computing Belongs Within the Social Sciences, CACM 63 / 8, 2020

Denning, P., Comer, D., Gries, D., Mulder, M., Tucker, A., Turner, J., and PAUL R. Young, P. (1989). Computing as a Discipline. CACM, 32/2. 1989

Denning, P., Martell, C. (2015) Great Principles of Computing. The MIT Press. 2015.

Forsythe, G. (1961). Engineering students must learn both computing and mathematics. J. Eng. Educ. 52, 1961.

Gibbs, E.B., Tucker, A.B. (1986). A model curriculum for a liberal arts degree in computer science. CACM, 29/ 3. 1986.

Haigh, T. (2023): Conjoined Twins: Artificial Intelligence and the Invention of Computer Science. CACM 66/6. 2023.

Haigh, T. (2025): Artificial Intelligence Then and Now. CACM, 68/2. 2025.

Hevner, A. R., March, S. T., Park, J., Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28/1. 2004

Knuth, D,E. (1974). Computer programming as an art. ACM Turing award lectures. https://doi.org/10.1145/1283920.1283929

Knuth, D.E (1974). Computer Science and Its Relation to Mathematics; The American Mathematical Monthly, 81 / 4. (1974),

Knuth D.E. (2023) The Art of Computer Programming, Volumes 1-4B. Reading, Massachusetts: Addison-Wesley, 2023

McCarthy, J. (2007). What is Artificial Intelligence, [online]. Stanford University. Available, 2007. at: http://www-formal.stanford.edu/jmc/whatisai.pdf.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. Hanover, NH: Dartmouth College.

Minsky, M. L. (Ed.). (1968). Semantic Information Processing. MIT Press. (1968)

Noy, S., Zhang, L. (2023). Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence (Working Paper). Available at SSRN: https://ssrn.com/abstract=4375283 or http://dx.doi.org/10.2139/ssrn.4375283. 2023.

Nygaard, K. (1986): Program Development as a Social Activity In: Kugler, H.J. (ed.) Information Processing 86, Elsevier Science, IFIP. Proceedings from the IFIP 10th World Computer Congress, Dublin, Ireland, September, 1986.

Shojaee, P., Mirzadeh, I., Alizadeh, A., Horton, M., Bengio, S, Farajtabar,M. (2025). The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. arXiv preprint arXiv:2506.06941, 2025

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A. (2006). A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. ACM, New York, New York, October, 2003. (2nd ed., 2006)

Turing. A.M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society 42.3. 1936.

Vardi, M. (2024). Is Computing a Discipline in Crisis? CACM, 67/71. 2024.

von Neumann, J. (1945): First draft report on the EDVAC. Moore School of Computing. Univ Pennsylvania (for the US Army Ordnance Department), 1945.

von Neumann, J. (1966) Theory of Self-Reproducing Automata Author: (completed and edited by Arthur W. Burks). University of Illinois Press. 1966.

Werthner, H. (2025). Digital Humanism – On Digitalization and Artificial Intelligence. Springer, 2025.