Why Has AI Failed?
And How Can it Succeed?

John F. Sowa

Abstract. In the 1960s, pioneers in artificial intelligence made grand claims that AI systems would surpass human intelligence before the end of the 20th century. Except for beating the world chess champion in 1997, none of the other predictions have come true. But AI research has contributed a huge amount of valuable technology, which has proved to be successful on narrow, specialized problems. Unfortunately, the field of AI has fragmented into those narrow specialties. Many researchers claim that their specialty is the key to solving all the problems. But the true key to AI is the knowledge that there is no key. Human intelligence comprises every specialty that anyone in any culture or civilization has ever dreamed of. Each one is adequate for a narrow range of applications. The power of human intelligence comes from the ability to relate, combine, and build on an open-ended variety of methods for different applications. Successful AI systems require a framework that can support any and all such combinations.

This article, which was presented in a keynote talk at the Mexican International Conference on AI, November 2014, is a slightly revised preprint of a version in *Computación y Sistemas,* Vol. 18 No. 3, 2014, pp. 1-5. For the slides, see http://www.jfsowa.com/talks/micai.pdf

1. Early Success and Later Disappointment

From 1945 to 1970, computer hardware, software, and theory developed rapidly. By the 1960s, the prospects for artificial intelligence seemed unlimited. From a textbook on machine translation in 1960: "While a great deal remains to be done, it can be stated without hesitation that the essential has already been accomplished" [2]. In 1965, I. J. Good defined an ultraintelligent machine as a system that surpasses human ability in every field [4]. He predicted "It is more probable than not that, within the twentieth century, an ultraintelligent machine will be built and that it will be the last invention that man need make." Marvin Minsky was a technical adviser for the AI features of the HAL 9000 computer in the movie *2001: A Space Odyssey.* When the movie opened in 1968, he claimed that it was a "conservative estimate" of AI technology in 2001.

Those claims seemed reasonable at the time. By the early 1960s, research on AI had produced an impressive body of results. Many of them were documented in the book *Computers and Thought* [3]:

- In the first paper, Alan Turing asked the fundamental question of AI, "Can a machine think?" As a test for thinking, he proposed an *imitation game*: if people cannot distinguish a computer's responses from human responses in a dialog, then they should consider the computer to be thinking at a human level [22]. Today's consensus is that pattern-matching methods enable a computer to imitate human responses in a short conversation, but not in complex interactions.

- Arthur Samuel invented machine learning methods that are still widely used today. He demonstrated their power in a checker-playing program that learned to play the game better than its designer. It even beat the Connecticut state champion [18].

- Other projects developed methods for answering questions in English, proving theorems in geometry, solving calculus problems, and managing an investment portfolio. With many variations and extensions, the methods are still the basis for the AI toolkit today: list processing, pattern matching, grammars, logic, heuristics, statistics, and neural networks.

- But some papers made questionable claims about simulating human thought, concept formation, or language understanding. As in the Turing test, a successful imitation of short examples is not convincing evidence of deeper thought or understanding.

For machine translation (MT), the technology of the 1960s produced useful results [8,9]. At the 1964 World's Fair, IBM demonstrated a system that translated Russian to English and printed both languages with the interchangeable bouncing balls of their Selectric typewriters. A competing system, the Georgetown University Automatic Translator (GAT), was widely used for translating physics research from Russian to English [6]. In the 1970s, an upgraded version of GAT was commercialized as SYSTRAN, which is still a widely used MT system. Today, free translators are available for any page on the WWW. But by the criterion of Fully Automatic High-Quality Translation (FAHQT), professional human translation is still far superior.

The predictions by Good and Minsky were wrong because the time scale they considered was much too short. The exponential growth in hardware speed and capacity enabled a supercomputer to beat the world chess champion in 1997 [7]. But improvements in software theory and practice did not keep pace. From a historical perspective, the seemingly rapid growth in early computer science took advantage of many centuries of prior research:

- Aristotle established the foundations for formal logic, ontology, and cognitive science. His theory of language was actively debated for centuries, and modern linguists have adopted aspects of his ontology [1]. Philosophers claim that his psychology provides better insights than 20th century behaviorism [16]. Even today, the RDF(S) notation for the Semantic Web does not use any logic that goes beyond Aristotle's syllogisms. OWL is more expressive, but many of the published ontologies use only the Aristotelian subset [20].

- Diagrams and mechanical aids for reasoning and computation have been used since antiquity. Euclid drew elaborate diagrams as essential components of proofs. In the 3rd c AD, Porphyry included a tree diagram in his introduction to Aristotle's categories. In the 14th c, Ramon Lull combined the tree of Porphyry with rotating circles as a method for relating categories to generate new ones. After studying Lull's rotating circles, Leibniz developed an equivalent numeric method: map primitive concepts to prime numbers; multiply the primes to represent compound concepts; and use division to test whether one concept subsumes another. To automate the arithmetic, he invented the first mechanical calculator for multiplication and division. Leibniz's method inspired Gödel to map all logical formulas to products of primes for the proof of his famous theorem. Leibniz also invented binary arithmetic, which Boole adopted for his logic and which Turing adopted for his machines, both abstract and electronic.

- Medieval logic put more emphasis on semantics than syntax. In his *Summa Logicae*, William of Ockham developed a *model-theoretic semantics* for a subset of Latin. He combined Aristotle's logic, the propositional logic of the Stoics, and even a subset of modal and temporal logic. Among the logicians who studied that logic were Bolzano, Brentano, Peirce, and the Polish school, which included Tarski. In 1887, Peirce published an article "Logical Machines" in the *American Journal of Psychology* [17]. He described Babbage's mechanical computers and machines for proving theorems in Boolean algebra. That paper was included in the bibliography of *Computers and Thought*.

By 1980, the legacy of the previous centuries had been translated to a computable form. Applications of expert systems, pattern recognition, logic programming, and natural language processing (NLP) showed promise of great things to come. In 1982, the Japanese launched the Fifth Generation project based on AI software and massively parallel hardware. But by the end of the 1980s, AI software did not scale to larger applications, and special-purpose hardware was less cost-efficient than the mass-produced microprocessors. The early predictions seemed unlikely, research funding dried up, and the field went into an "AI winter" [5].

2. Future Directions

Since the 1990s, the huge volumes of data on the Internet made AI methods of deep reasoning and language analysis impractical. The research shifted to shallow statistical methods for information retrieval, data mining, and machine translation. The two most impressive successes for deeper methods used supercomputers. In 1997, the Deep Blue system beat Gary Kasparov, the world chess champion [7]. In 2011, the Watson system beat two of the best Jeopardy! contestants [10,13]. But cynics claimed that the main purpose of those projects was advertising for IBM computers.

Whatever the motivation, the chess system demonstrated the importance of hardware speed and capacity. But it did little to advance AI research. The Watson system, however, showed how a combination of language analysis, reasoning methods, machine learning, and large volumes of data could match human performance on challenging problems. With further improvements in hardware and software, a version of Watson running on more economical servers is being used to diagnose cancer and other diseases [11]. It doesn't replace physicians, but it gives them better advice and more focused information than search engines.

Despite some impressive applications, no AI system today can learn, understand, and use language as quickly and accurately as a 3-year-old child. Automated telephone systems are useless when the caller strays from a preprogrammed script. Computer help facilities are useless when the user doesn't know or can't remember the exact name of the command, feature, or menu item. To be successful, AI systems don't have to as intelligent as the HAL 9000. But they need to be flexible, adaptable, helpful, and able to communicate in the user's native language. For specialized applications, they should be at least as advanced as Watson. But they should be able to learn those applications by reading books and asking questions. Whether they pass the Turing test is irrelevant.

The AI technology developed in the past 60 years is sufficient to support such systems. No major breakthroughs were necessary to implement Watson. It was assembled in a few years by putting together readily available AI components. Its English parser, for example, is based on a 20-year-old design [13], but the version used in Watson is faster and more accurate than more recent statistical parsers. A special-purpose pattern matcher was designed for Watson, but it turned out to be slower and less general than the Prolog language, which is over 30 years old [10]. But Watson required a great deal of applied research to tailor the components, make them work together, and test the many combinations on typical Jeopardy! questions. Unfortunately, few programmers and system analysts have the expertise to design and maintain such systems.

The great strength of AI technology is its coverage of nearly every aspect of intelligence. But its great weakness is fragmentation. Researchers who specialize in any area try to make their favorite set of tools do everything. Logicians combine formal logics with formal ontologies, formal grammars, and formal methods for mapping one to another. Specialists in neural networks try to solve every problem by combining multiple networks to form deeper networks. The strength of Watson is that it combines multiple modules based on different paradigms. But those modules are lashed together with procedural code. To manage that diversity, Minsky [14] proposed a "society" of active processes:

> What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single, perfect principle. Our species has evolved many effective although imperfect methods, and each of us individually develops more on our own. Eventually, very few of our actions and decisions come to depend on any single mechanism. Instead, they emerge from conflicts and negotiations among societies of processes that constantly challenge one another. (§30.8)

This view is radically different from the assumption of a unified formal logic that cannot tolerate a single inconsistency. Minsky's goal is to build a flexible, fault-tolerant system. To provide the motivation that drives an intelligent system, Minsky extended his *Society of Mind* with an *Emotion Machine* [15]. But much more detail is needed to specify how the processes can and should interact in an efficient computer implementation.

As an architecture that can support a society of interacting agents, Sowa designed the Flexible Modular Framework (FMF), which enables heterogeneous processes to interact by passing messages in various languages [19]. Majumdar used the FMF to support a hierarchy of agents that behave like the managers and employees of a business [12]. The chief executive officer (CEO) gives the organization a coherent "personality" for external interactions. Reporting to the CEO is a hierarchy of vice presidents in charge of major divisions, directors of important functions, lower-level managers, and specialists that perform an open-ended variety of cognitive tasks. For an overview of these and other promising designs, see the article "Future directions in semantic systems" [21].

References

1. Arens, Hans (1984) *Aristotle's Theory of Language and its Tradition*, Amsterdam: John Benjamins Publishing Co.
2. Delavenay, Emile (1960) *An Introduction to Machine Translation*, London: Thames & Hudson.
3. Feigenbaum, Edward A., & Julian Feldman, eds. (1963) *Computers and Thought*, New York: McGraw-Hill.
4. Good, Irving John (1965) Speculations concerning the first ultraintelligent machine, in F. L. Alt & M. Rubinoff, eds., *Advances in Computers* **6**, New York: Academic Press.
5. Hendler, James A. (2008) Avoiding another AI winter, *IEEE Intelligent Systems* **23:2**, 2-4.
6. Henisz-Dostert, Bożena, R. Ross Macdonald, & Michael Zarechnak (1979) *Machine Translation*, The Hague: Mouton.
7. Hsu, Feng-hsiung (2002) *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*, Princeton University Press.
8. Hutchins, W. John (1986) *Machine Translation: Past, Present, Future*, Ellis Horwood, Chichester, England.
9. Hutchins, W. John (2015) Machine translation: History of research and applications, to appear as chapter 6 in *The Encyclopedia of Translation Technology*, ed. Chan Sin-wai, London: Routledge.
10. Lally, A., J. M. Prager, M. C. McCord, B. K. Boguraev, S. Patwardhan, J. Fan, P. Fodor, J. Chu-Carroll (2012) Question analysis: How Watson reads a clue, *IBM Journal of Research & Development* **56:3/4**. http://www.patwardhans.net/papers/LallyEtAl12.pdf
11. Leske, Nicola (2014) Doctors seek help on cancer treatment from IBM supercomputer, *Reuters*, http://in.reuters.com/article/2013/02/08/ibm-watson-cancer-idINDEE9170G120130208
12. Majumdar, Arun K., & John F. Sowa (2009) Two paradigms are better than one, and multiple paradigms are even better, in S. Rudolph, F. Dau, & S. O. Kuznetsov, eds., LNAI 5662, Berlin: Springer, pp. 32–47. http://www.jfsowa.com/pubs/paradigm.pdf

13. McCord, Michael C., J. W. Murdock, & B. K. Boguraev (2012) Deep parsing in Watson, *IBM J. of Research & Development* **56:3/4**. http://brenocon.com/watson_special_issue/03%20Deep%20parsing.pdf

14. Minsky, Marvin (1986) *The Society of Mind*, New York: Simon & Schuster.

15. Minsky, Marvin (2006) *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*, New York: Simon & Schuster.

16. Nussbaum, Martha C., & Hilary Putnam (1992) Changing Aristotle's mind, in M. C. Nussbaum & A. O. Rorty, eds., *Essays on Aristotle's De Anima*, Oxford: University Press, pp. 27-56. Reprinted in H. Putnam, *Words and Life*, Cambridge, MA: Harvard University Press, pp. 22-61.

17. Peirce, Charles Sanders (1887) Logical machines, *American Journal of Psychology* **1**, Nov. 1887, pp. 165-170.

18. Samuel, Arthur L. (1959) Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development* **3**, 211-229.

19. Sowa, John F. (2002) Architectures for intelligent systems, *IBM Systems Journal* **41:3**, 331-349. http://www.jfsowa.com/pubs/arch.htm

20. Sowa, John F. (2007) Fads and fallacies about logic, *IEEE Intelligent Systems*, **22:2**, 84-87. http://www.jfsowa.com/pubs/fflogic.pdf

21. Sowa, John F. (2011) Future directions for semantic systems, in A. Tolk & L. C. Jain, eds., *Intelligence-Based Systems Engineering*, ISRL 10, Berlin: Springer, pp. 24-47. http://www.jfsowa.com/pubs/futures.pdf

22. Turing, Alan M. (1950) Computing machinery and intelligence, *Mind* **49**, 433-480.