From Alan Turing to modern AI: practical solutions and an implicit epistemic stance

Article *in* AI & Society · February 2016

DOI: 10.1007/s00146-016-0646-7

2 authors:

George F. Luger
University of New Mexico
111 PUBLICATIONS **2,121** CITATIONS
SEE PROFILE

Chayan Chakrabarti
University of New Mexico
15 PUBLICATIONS **42** CITATIONS
SEE PROFILE

AI and Society manuscript No.
(will be inserted by the editor)

From Alan Turing to Modern AI: Practical Solutions and an Implicit Epistemic Stance

George F. Luger · Chayan Chakrabarti

Received: date / Accepted: date

Abstract It has been just over 100 years since the birth of Alan Turing and more than sixty-five years since he published in *Mind* his seminal paper, *Computing Machinery and Intelligence* [78]. In the Mind paper, Turing asked a number of questions, including whether computers could ever be said to have the power of "thinking"[1]. Turing also set up a number of criteria - including his *imitation game* - under which a human could judge whether a computer could be said to be "intelligent". Turing's paper, as well as his important mathematical and computational insights of the 1930s and 1940s led to his popular acclaim as the "Father of Artificial Intelligence". In the years since his paper was published, however, no computational system has fully satisfied Turing's challenge.

In this paper we focus on a different question, ignored in, but inspired by Turing's work: How might the Artificial Intelligence practitioner implement "intelligence" on a computational device? Over the past 60 years, although the AI community has not produced a general-purpose computational intelligence, it has constructed a large number of important artifacts, as well as taken several philosophical stances able to shed light on the nature and implementation of intelligence.

G. Luger
Computer Science Department
University of New Mexico
Albuquerque NM
E-mail: luger@cs.unm.edu

C. Chakrabarti
Computer Science Department
University of New Mexico
Albuquerque NM
E-mail: cc@cs.unm.edu

[1] *"I propose to consider the question, Can computers think?" ...*
Alan Turing, Computing Machinery and Intelligence, Mind, 1950.

This paper contends that the construction of any human artifact includes an implicit epistemic stance. In AI this stance is found in commitments to particular knowledge representations and search strategies that lead to a product's successes as well as its limitations. Finally, we suggest that *computational* and *human* intelligence are two different natural kinds, in the philosophical sense, and elaborate on this point in the conclusion.

Keywords Artificial Intelligence · Computational Intelligence · Epistemic Stance

1 Introduction: The Imitation Game

Turing proposed to answer the "Can computers think" question by introducing a gedanken experiment called the *imitation* game [24, 78]. In the imitation game a human, the "interrogator", asks questions of two different entities, one a human and the other a computer. The interrogator is isolated from the two respondents so that he/she does not know whether the human or computer is answering. Turing, in the language of the 1940s, comments that "the ideal arrangement is to have a teleprinter communicating between the two rooms", ensuring the anonymity of the responses. The task of the interrogator is to determine whether he/she is communicating with the computer or the human at any time during the question answering session. If the interrogator is unable to determine, on average, whether the human or the machine is responding to questions, Turing contends that the computer has to be seen as "thinking", or, more directly, to possess intelligence.

The historical timing of Turing's paper is very instructive. It appeared before computers were challenged to understand natural human languages, play expert level chess, recognize visual scenes, or control robots in deep space. Turing and others [77, 15, 64] had already formally specified what it *meant to compute*, and had by that time hypothesized limits on *what was computable*. This *sufficient* model for any computation is often called the Church/Turing hypothesis [77]. However, the radio-tube-based-behemoths of Turing's time were used mainly to work out the trajectories of ordnance and to break complex ciphers. It is important to realize then - given the very limited nature of actual tasks addressed at that time by computers - that the most important result of Turing's imitation game was to challenge humans to consider whether or not thinking and intelligence are uniquely human skills. The task of Turing's imitation game was an important attempt to separate the attributed skills of "thinking" and "intelligence" from their human embodiment.

Of course, no informed critic would contend that electronic computers, at least as presently configured, are *universally* intelligent - they simply do a large number of specific but complex tasks - delivering medical recommendations, guiding surgeries, playing chess or backgammon, learning relationships in large quantities of data, and so on - as well as, and often much better than, their human counterparts performing these same tasks. In these limited situations,

computers have passed Turing's test. Facetiously, with the power and scope of current computation, it could also be said that humans have failed Turing's test for machine-based intelligence.

It is interesting to note, also, that many in the research community are still trying to play/win this challenge of building a general purpose intelligence that can pass the Turing test in any area where a human might challenge it. This can be seen as useful, of course, for it requires the computer and program designer to address the more complete and complex notion of building a general-purpose intelligence. Perhaps the program closest to achieving this goal is IBM's Watson, the winner of the Jeopardy television challenge of February 2011, see Wikipedia Watson Computer [25]. Commercially available programs addressing the quest for general intelligence include web chat bots, such as Apple's Siri. The Turing challenge remains an annual event, and the interested reader may visit Wikipedia at Turing Test Loebner Prize, for details.

In fact, the AI community often uses forms of the imitation game to test whether their programs are ready for actual use. When the computer scientists and medical faculty at Stanford were ready to deploy their MYCIN program they tested it against a set of outside medical experts skilled in the diagnosis of meningitis infections [7]. The results of this analysis were very interesting, not just because, in the double-blind evaluation, the MYCIN program out performed the human experts, but also because of the lack of a general consensus - only about 70% agreement - on how the human experts themselves would treat these patients. Besides evaluating many deployed expert systems, a form of Turing's test is often used for testing AI-based video games, chess and backgammon programs, computers that understand human languages, and various forms of web agents [75].

The failure, however, of computers to succeed at the task of creating a general-purpose thinking machine begins to shed some understanding on the "failures" of the imitation game itself. Specifically, the imitation game (and Turing's paper) offer little hint of a definition of intelligent activity nor do they offer specifications for building intelligent artifacts. Some critics contend that Turing chose to avoid trying to offer a specific definition of intelligence [24]. Deeper issues remain that Turing did not address: What IS intelligence? What IS grounding or how may a human's or a computer's statements be said to have "meaning"? Finally, can humans understand their own intelligence in a manner sufficient to formalize or replicate aspects of it on a computer?

Perhaps the greatest contribution of artificial intelligence (and, in general, of computation) to the study of knowledge and human cognitive processing is to cash-out Ryle's "ghost in the machine" [68]. Artificially intelligent artifacts offer a direct challenge to mind/body dualism. Many (mostly rationalist) philosophers, including Descartes, Leibnitz and Spinoza contend that a non-material substance, e.g., Descartes' *res cogitans* [20], is required (is necessary) to support human intellection. Computation has demonstrated an alternative: The mind/brain is a processor, composed of billions of sub-processors, and the resulting product, whether thought, feeling, or action, is an artifact of this

processing. It follows that the human mental world can be grounded in a physical embodiment, can generate an infinite range of emotions and actions with a finite instruction set, and that the mind itself is a complex system composed of many interacting components [55].

It remains the fact, however, that designing and building a program for a computer is a human intellectual activity, and requires sufficient answers to questions such as: How can I represent for the computer entities and processes occurring in this natural world? How do I understand and represent complex interacting processes? How can I measure success within the sub-constraints of a complex task? How do I capture teleology or goal-directed behavior? All these questions entail an implicit epistemic stance.

This paper considers these issues, especially the responses to the challenge of building intelligent artifacts that the artificial intelligence community has taken since Turing. In the next section we give a brief overview of several AI programs built over the past sixty years to be "intelligent" problem solvers. We see, often apart from the practical stance of the program's original designers, many of the earliest approaches to AI as having an a priori bias towards the empiricist or rationalist or pragmatist traditions for understanding an external world. In the third section we present a constructivist rapprochement that addresses many of the epistemic assumptions of early AI work. Finally, we offer some preliminary conjectures about how a Bayesian model might be epistemologically plausible.

2 AI Programs as Adventures in Rationalism, Empiricism, and Pragmatism

The study of epistemology considers how the human agent knows itself and its world, and, in particular, whether this agent/world interaction can be considered as a topic for scientific study. The empiricist, rationalist, and pragmatist traditions have offered their differing answers to this question and artificial intelligence researchers have made these approaches concrete with their programs. We are not suggesting that individual AI program designers ARE rationalists, empiricists, or pragmatists, rather that their approaches to problem solving can be understood from these various perspectives. It is only natural that a discipline that as its focus engages in the design and building of artifacts that are intended to capture intelligent activity would intersect with philosophy and psychology, and in particular, with epistemology. We describe this intersection of disciplines in due course, but first we look at these philosophical traditions themselves.

Rationalism may be described as that philosophical position where, in the acquisition and justification of knowledge, there is a bias toward utilization of unaided reason over sense experience [1]. Clear and distinct ideas become a reality in themselves, and the *sine qua non* of mathematics and science. Perhaps the most influential rationalist philosophers after Plato were Leibnitz, Spinoza, and Descartes, central figures in the development of modern concepts

of the origins of thought and theories of mind. Descartes attempted to find a basis for understanding himself and the world purely through introspection and reflection. Descartes [20] systematically rejected the validity of the input of his senses and even questioned whether his perception of the physical world was "trustworthy". Descartes was left with only the "reality" of thought: the reality of his own physical existence could be reestablished only after making his fundamental assumption: "Cogito ergo sum". Establishing his own existence as a thinking entity, Descartes inferred the existence of a God as an essential creator and sustainer. Finally, the reality of the physical universe was the necessary creation and its comprehension was enabled through a veridical trust in this benign God.

Descartes' powers of abstraction and emphasis of clear and distinct ideas (the same powers that produced his mind/body dualism) offered excellent support for his creation of mathematical systems including analytic geometry, where mathematical relationships could provide the constraints for characterizing the physical world. It was a natural next step for Newton to describe Kepler's laws of planetary motion in the language of elliptical relationships of distances and masses. Descartes' clear and distinct ideas themselves became a sine qua non for understanding and describing "the real". His physical (res extensa) non-physical (res cogitans) dualism supports the body/soul or mind/matter biases of much of our own modern life, literature, and religion (How else might we interpret *the spirit is willing but the flesh is weak*).

The origins of many of the rationalists' ideas, especially the primacy of abstraction, can be traced back at least to Plato [4]. The epistemology of Plato supposed that as humans experience life through space and time we gradually come to understand the pure forms of real life separated from material constraints. In his philosophy of reincarnation, the human soul is made to forget its knowledge of truth and perfect forms when reborn into a new existence. As life progresses, the human, through experience, gradually comes to "remember" the forms of the disembodied life: learning is remembering. In his cave experience, in book seven of The Republic, Plato introduces his reader to these pure forms, the perfect sphere, beauty, and truth.

Mind/body dualism is a very attractive exercise in abstraction, especially for agents confined to a physical embodiment and limited by senses that can mislead, confuse, and even fail. This same rationalist power of abstraction is also at the core of computation. How else can sets of symbols or patterns of a process represent "something else", whether the trajectory of ordnance or patterns of human speech? Rationalism's embodiment entering the AI-age can be found in the early twentieth century analytic philosophers, the symbol-based AI practitioner Herb Simon [74], and especially in the works of the linguist Noam Chomsky [14]. It provides a natural starting point for work in AI as we see subsequently.

Empiricism may be described as that philosophical position that links all knowledge to experience. It often takes the form of denying that there is any a priori knowledge, or any knowledge necessary truths, or any innate knowledge supporting general principles [1]. Aristotle was arguably one of the first

proponents of the empiricist tradition, although his philosophy also contained the ideas of "form" and the ability to "abstract" from a purely material existence. However, Aristotle rejected Plato's doctrine of transcendent forms, noting that the act of abstraction does not entail an independent existence for the abstraction. For Aristotle the most important aspect of nature is change. In his Physics, he defines his "philosophy of nature" as the "study of things that change". He distinguishes the *matter* from the *form* of things: a sculpture might be "understood" as the material bronze taking on the form of a specific human. Change occurs when the bronze takes on another form. This matter/form distinction also supports the computer scientists' notions of symbolic computing and data generalization, where sets of symbols can *represent* entities in a world and abstract relations and algorithms describe how these entities can share common characteristics, as well as be systematically altered. Abstracting form from a particular material existence supports computation, the manipulation of abstractions, as well as theories for data structures and languages as symbol-based representations.

In the world of the enlightenment, the empiricist tradition of Locke, the early Berkeley, and Hume, distrusting the abstractions of the rational agent, remind us that nothing comes into the mind or to understanding except by passing through the sense organs of the agent. On this view the rationalist's perfect sphere, or absolute truth, simply do not exist. Locke suggests that the human at birth is tabula rasa, a blank slate, where all language and human "meaning" is captured as conditioning across time and experience. What the human agent "internalizes" are the human-perceptible aspects of a physical existence; what it "knows" are loose associations of these physical stimuli. The extremes of this tradition, expressed through the Scots philosopher David Hume, include a denial of causality and the ability to prove the existence of an all-powerful God. There is an important distinction here, the foundation of an agnostic/skeptic position: it is not that a God doesn't/can't exist, it is rather that the human agent can't know or prove that He/She does exist.

The empiricist tradition was especially strong in the first half of the twentieth century leading into the AI movement, where its supporters included A. J. Ayer and Rudolph Carnap, proponents of *logical empiricism*, who tried to fuse empiricism with a logic-based rationalism, as well as the behaviorist psychologist B. F. Skinner.

Pragmatism, as proposed by Peirce [60] and James [38], suggests that the meaning of a doctrine is the same as the practical effects of adopting it and contend that beliefs are true if they work satisfactorily in the widest sense of the word [1]. Whereas empiricism and rationalism can be seen as self-based characterizations of knowing, particularly as epistemology seems to be the product of internalized thought experiments, pragmatism asks what an action or stance will "effect" or "do" in a specific world environment. In short, pragmatism asserts meaning, as well as an ethical valence, to a word or action as it is externalized in an active world.

In Pragmatism [37], James asserts that "27" may mean *one dollar too few* or equally *a board one inch too long*. He asserts, "What shall we call a

thing anyhow? It seems quite arbitrary, for we carve out everything, just as we carve out constellations, to suit our human purposes". Further James claims "We break the flux of sensible reality into things, then, at our will. We create the subjects of our true as well as of our false propositions. We create our predicates also. Many of the predicates of things express only the relations of the things to us and to our feelings. Such predicates, of course are human additions".

Pragmatism, then, purports to ground all thoughts, words, and actions in their expected consequences. An example of this epistemological stance, from James, in the Varieties of Religious Experience [38] is that the *truth*, as well as any imputed *value*, of a particular religious stance is what that stance does for an individual's life, for example, help deal with an addiction problem or encourage the performance of charitable acts. This form of pragmatism allows little critique, however, as one person's religious values can directly contradict those of others, for instance with various "inquisitions" or "fundamentalist actions" all justified in the name of some religion. An important consequence of the pragmatist philosophy was John Dewey [21], a colleague of James and Peirce, who had an important impact on twentieth century education both in the US and worldwide.

A computer program designer/builder makes specific assumptions about his/her application domain, including: What will a program's variables represent? How will data relationships be captured? What strategies will support control algorithms? What is the relationship between the "perfect" and the "good enough" solution? These questions must be addressed in the program design and creation through the selection of specific software tools, for example, the explicit separation of domain knowledge/logic from control algorithms, as is common in building expert systems. These choices may be seen as a program builder's *epistemic stance* or *inductive bias* about an application domain. Many modern artificial intelligence practitioners have implicitly adopted empiricist, rationalist, and/or pragmatist views of the world. We conclude this section with several examples of each tradition.

From the rationalist perspective came the expert system technology where knowledge was seen as a set of clear and distinct relationships (expressed in *if/then* or *condition/action* rules) encoded within a production system architecture that could then be used to compute decisions in particular situations. In fact, these systems are often seen as extremely brittle, for example when an application situation does not exactly fit the logic specification, and too often, the human user is expected to be the interpreter for the interpreter.

Figure 1 offers a simplified example of the production system approach, where a rule set - the content of the production memory - is interpreted by the production system. When the *if* component of the rule is matched by the data in the working memory, the rule is said to "fire" and its conclusion then changes the content of the working memory preparing it for the next iteration of the system. The reader can observe that when the system is run in this "data-driven" mode it is equivalent to a modus ponens interpreter of *if/then* rule relationships.

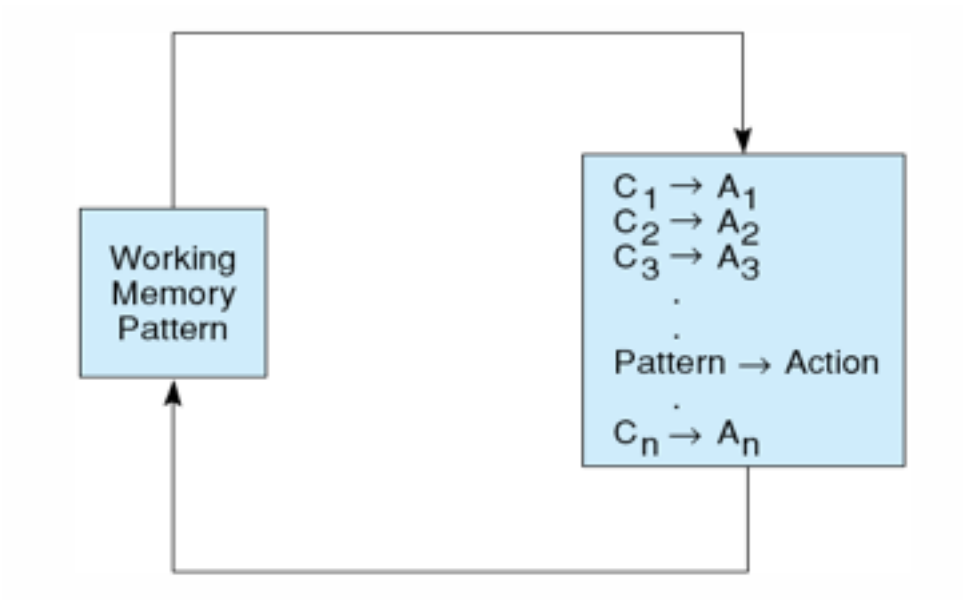


Fig. 1: A production system; in the traditional data-driven mode, a pattern in the Working Memory matches the condition of a rule in the Production Memory. When this occurs, the action of that rule takes place, producing new information for Working Memory and the system continues to iterate towards a solution.

Interestingly enough, when the same production system is run in goal-driven mode it can be seen as an abductive [60] interpreter. In this situation the goals we wish to solve - the explanations we want to prove "best" - are contained in the working memory and the production system takes these goals and matches them to the conclusions, the action or *then* components of the rules. When a conclusion is matched the rule again "fires" and the system puts the *if* pattern of the rule into the working memory to serve as a subgoal for the next iteration of the system, matching the conclusions of new rules. In this abductive mode the system searches back through a sequence of subgoals to see if it can make the case for the original goal/explanation to be true. Abduction is an unsound form of reasoning, so the abductive interpreter can be seen as generating possible explanations for the data. In many cases, some probabilistic or certainty factor measure is included with each rule supporting the interpreter's likelihood of producing the "best" explanation.

In the work of Newell and Simon [57] and Simon [74] this production system interpreter was taken a further step towards cognitive plausibility. On the Newell and Simon view, the production memory of the production system was a characterization of the human long-term memory and the if/then rules were seen to encode specific components of human knowledge. On this approach, human expertise for the practicing physician or the master chess player, for example, was acknowledged to be about 50,000 such rules [57]. The working

memory of the production system was seen as the human's short-term memory, or as describing a "focus of attention" for what the human agent was considering at any specific time (neuroscientists now localize this component of human processing in Broadmann's areas of pre-frontal cortex). Thus the production system was proposed as a *cognitive architecture* that took the current focus of the agent and used that to "fire" specific components of knowledge (rules) residing in long-term memory, which, in turn, changed the agent's focus of attention. Furthermore, production system learning [67] was seen as a set of procedures that could encode an agent's repeated experiences in a domain into new if/then rules in long-term (production) memory. The production system is often seen as an embodiment of Newell and Simon's *physical symbol system hypothesis* [57] described in their 1976 Turing award lecture and discussed further in the conclusion.

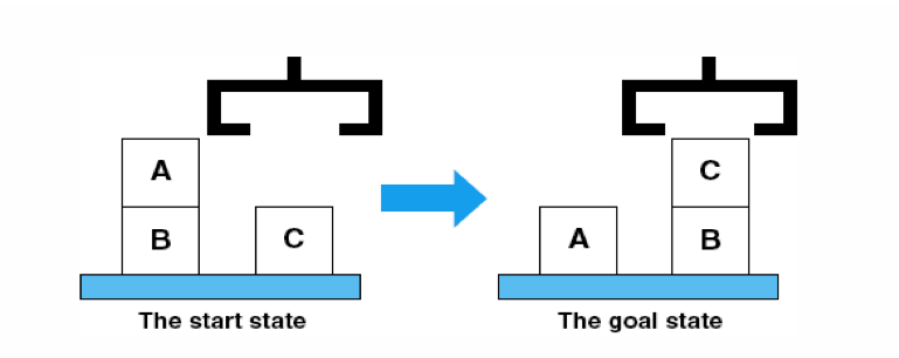
Early design of robot systems [26] can also be seen as a rationalist exercise where the world is described as a set of explicit constraints that are organized as "states" with operators used to generate new states to accomplish a particular task. "States" of the world are represented as a set of predicate calculus descriptions and then these are checked by a set of "move" rules that are used to generate new states of the world, much as the production system did in the previous example. Figure 2a presents a start state and a goal state for a configuration of blocks and a robot arm. These states are then changed by applying "move" predicates, as can be seen in the state space of Figure 2b. Problems can happen, of course, when the actual world situation is not represented precisely as the logic specifications would suggest, e.g., when one block or the robot arm accidentally moves another.

Several later approaches to the design of control systems take a similar approach. When NASA designed a planning system for controlling the combustion systems for deep space vehicles, it expressed the constraints of the propulsion system as sets of propositional calculus formulae. When the control system for the space vehicle detected any anomaly it searched these constraints to determine what to do next. This system, NASA's Livingstone, proved very successful for guiding the space flight in deep-space situations [81, 82].

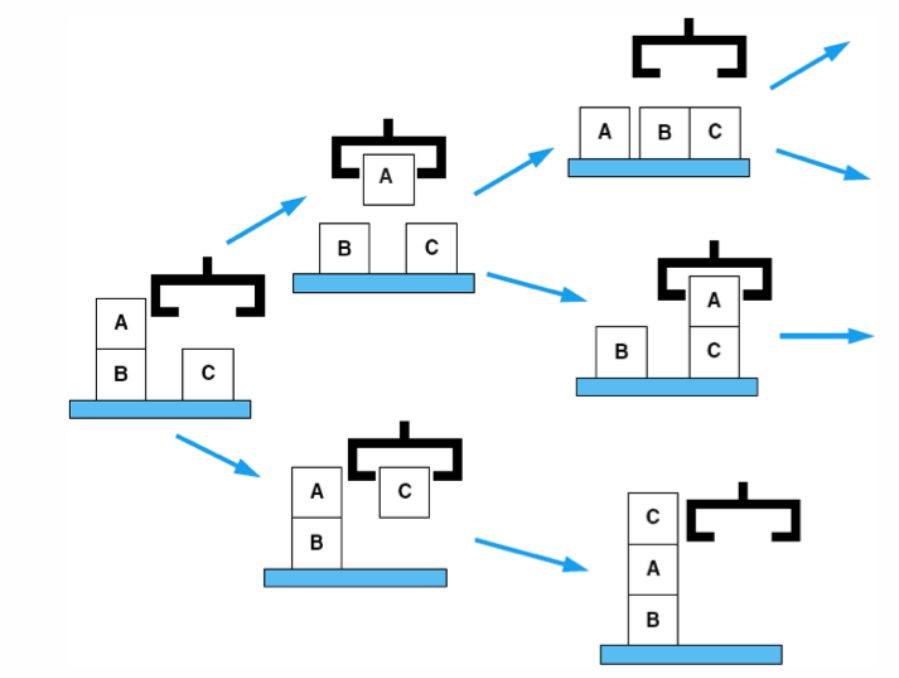
There are many other examples of this rationalist bias in AI problem solvers. For example, case-based reasoning uses a data base of collected and clearly specified problem solving situations, much as a lawyer might look for earlier legal precedents, cases that can be modified and reused to address new and related problems [42].

A final example of the rationalist perspective is the suggestion that various forms of logic in representation and inference, can be sufficient for capturing intelligent behavior [47, 50, 43]. Many interesting and powerful representations have come from this work including non-monotonic logics, truth-maintenance systems, and assumptions of minimal models or circumscription [48, 49, 45].

From the empiricist view of AI there is the creation of semantic networks, conceptual dependencies, and related association-based representations. These structures, deliberately formed to capture the concept and knowledge associations of the human agent, were then applied to the tasks of understanding



(a)
start = [handempty, ontable(b), ontable(c), on(a,b), clear(c), clear(a)]
goal = [handempty, ontable(a), ontable(b), on(c,b), clear(a), clear(c)]



(b)
move(pickup(X), [handempty, clear(X), on(X,Y)], [del(handempty), del(clear(X)), del(on(X,Y)), add(clear(Y)), add(holding(X))])

Fig. 2: (a) The start and goal states of a blocks-world problem, and the set of predicate descriptions for each state. (b) presents part of the state space search representing the movement of the blocks to attain a goal state. The move procedure (stated as preconditions, add, and delete constraints on predicates) is one of many possible predicates for changing the current state of the world.

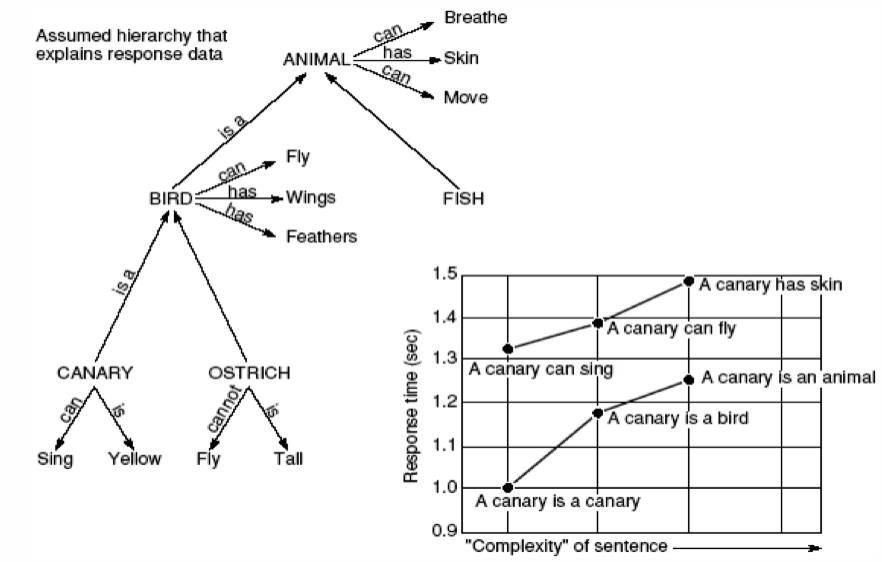


Fig. 3: A semantic net "bird" hierarchy (left) that is created from the reaction time data (right) of human subjects (Collins and Quillian 1969). This figure is adapted from [33].

human language and interpreting meaning in specific contexts. The original semantic networks were, in fact, taken from the results of psychologists' Collins and Quillian, [17] reaction-time experiments. The goal was to design associative networks for computer-based problem solvers that captured the associative components of actual human memory. In their reaction time experiments shown in Figure 3, Collins and Quillian [17] hypothesized that the longer the human subject took to respond to a query, for example, "Does a bird have skin?", the further "apart" these concepts were assumed to be in the human memory system. Closely associated concepts would support more immediate responses.

A number of early AI programs sought to capture this associative representation, first Quillian [65] himself, with the creation and use of semantic networks. Wilks [80] basing his research on earlier work by Masterman [46] who defined around 100 primitive concept types, also created semantic representations for the computer-based understanding of human language. Schank and Colby [71] with their *conceptual dependency* representation, created a set of association-based primitives intended to support language-based meaning to be used for computer understanding or translation. Finally, John Sowa [75] created a *conceptual graphs* language whose structures could be reduced to forms of first-order logic. This approach allows a transduction of the earlier associative network schemes to a more rationalist-based representation (that can also support alternative interpretations of semantic closeness).

From the empiricist perspective, neural networks and "deep" semantic networks were also designed to capture associations in collected sets of data and

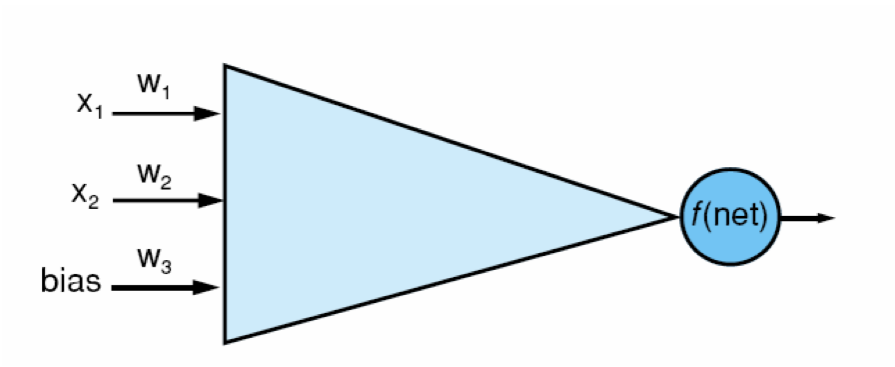
then, once trained, to interpret new related patterns in the world. For example, the back-propagation algorithm in training phase takes a number of related situations, perhaps surface patterns for an automated welder or phone patterns of human speech, and conditions a network until it achieves a high percentage of successful pattern recognition. Figure 4a presents a typical neuron from a back-propagation system. The input values for each neuron are multiplied by the (conditioned) weights for that value and then summed to determine whether the threshold for that neuron is reached. If the threshold is reached the neuron fires, usually generating an input signal for other neurons. The back-propagation algorithm, Figure 4b, differentially "punishes" those weights responsible for incorrect decisions. Over the time of training the appropriately configured and conditioned network comes to "learn" the perceptual cues that solve a task. And then the trained network can be used to solve new related tasks.

Back-propagation networks are an example of *supervised* learning, where appropriate rewards and/or punishments are used in the process of training a network. Other network learning can be *unsupervised* where algorithms classify input data into "clusters" either represented by a prototype pattern or by some "closeness" measure. New input patterns then enter into the basins of attraction offered by the currently clustered patterns. In fact, many successful families of networks have been created over the years. There have also been many obvious - and scientifically useless - claims that neural connectivity (networks) ARE the way humans performed these tasks, *and therefore* appropriate representations for use in computer-based pattern recognition.

In an interesting response to the earlier rationalist planners for robotics described above, Brooks at the MIT AI laboratory created what he called the "subsumption" architecture [5, 6]. The subsumption architecture was a layered collection of finite state machines where each level of the solver was constrained by layers below it. For example, a "wander" directive at one level of the robot's controller would be constrained by a lower level that prevented the agent from "running into" other objects during wandering.

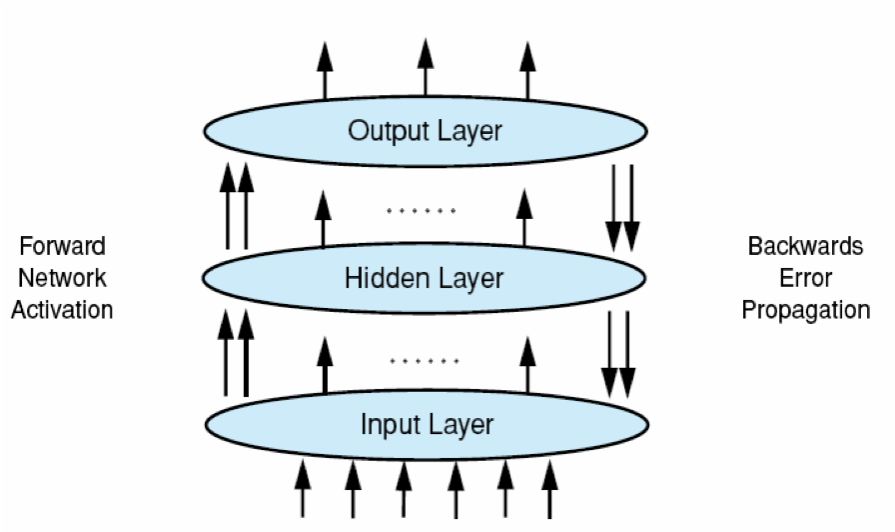
The subsumption architecture is the ultimate knowledge-free system in that there are no memory traces (states) ever created that could reflect situations that the robot had already learned through pattern association. Obviously such a system, although sufficient to explore its local environment, would not be able to find its way around a complex environment, for example, the roadways and alleys of a large city such as New York or Mumbai. Brooks acknowledged the fact of a memory free solver, in entitling his 1991 paper "Intelligence Without Representation" [6].

Other examples of representations with an empiricist bias include *artificial life* and *genetic algorithms*. These approaches, where information is usually encoded as bit-strings and whose operators include mutation and crossover [45], may be characterized as association and reward based solvers that are intended to capture survival of the fittest. Their advocates often saw these approaches as plausible models incorporating evolutionary pressures to produce emergent phenomena, including the intelligent behavior of an agent.



(a)

$$net = x_1w_1 + x_2w_2 + (bias)w_3$$



(b)

Fig. 4: (a) presents a single artificial neuron whose input values, multiplied by (trained) weights, produce a value, net. Usually using some sigmoid function, f(net), produces an output value that may, in turn, be an input for other neurons. (b) is a simple backpropagation network where input values move forward through the nodes of the network. During training the networks weights are differentially "punished" for incorrect responses to the input. Figures are adapted from [45]

Interesting examples of the pragmatist epistemological stance are programs meant to communicate with human agents. In the simplest sense, these programs do question answering, as for example, Apple's Siri or IBM's Watson. In a more demanding environment, programs are able to have a dialogue or a more complete conversation with a human user. Typical examples of this task might be when a user gets on-line to change a password or, more interestingly, to get financial, insurance, or hardware troubleshooting advice. In these sit-

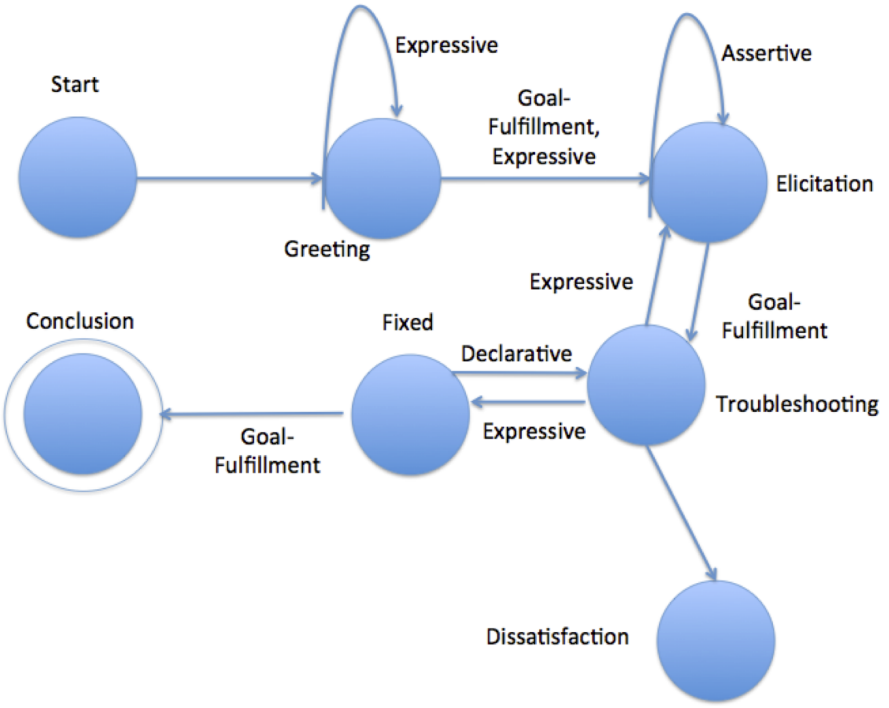


Fig. 5: A probabilistic finite state automaton for conversations in the "troubleshooting" domain, from [10]. Each state transition contains a probability measure specifying its likely use.

uations the responding program must have some notion of teleology, or the implicit purpose of the conversation.

Chakrabarti [8, 9, 10] has created such a system where probabilistic finite state machines, Figure 5, monitor whether the human agent's implied goal is met by the computational dialogue system. Figure 5 depicts a finite state machine representing a troubleshooting conversation, where the states are the components of the conversation $\{Start, Greeting, Elicitation, Troubleshooting, Fixed, Dissatisfaction, Conclusion\}$, and the transitions are speech acts and dialog acts $\{Expressive, Assertive, Declarative, Goal-Fulfillment\}$ [72, 73, 8, 10]. Meanwhile, at each step in the communication, a data structure, called a goal-fulfillment-map, Figure 6, captures the knowledge necessary to answer particular questions [8, 10].

This dialog management software demonstrates a method for combining content semantics, in the form of goal-fulfillment maps, within the pragmatic constraints of a conversation. A good conversation depends on both a goal-directed underlying process and a grounding in a set of facts about a knowledge domain. Chakrabarti's [8, 10] approach combines content semantics in the form of a rationalist-type knowledge engine with pragmatic semantics in the form a conversation engine to generate artificial conversations.

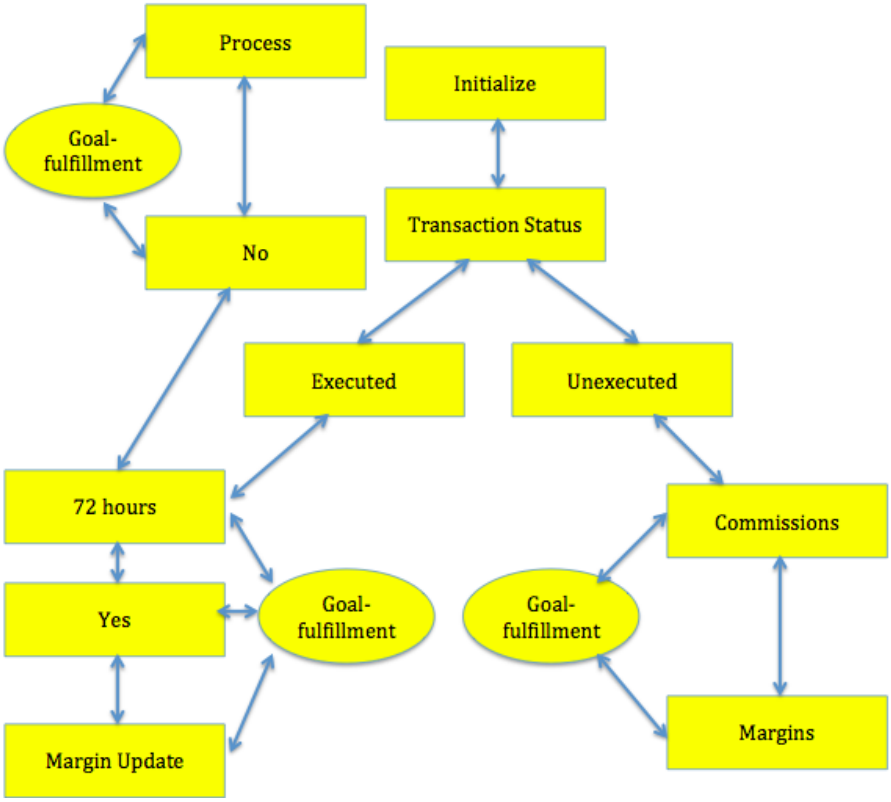


Fig. 6: A goal-fulfillment map that supports discussion of a financial transaction, from [10].

The knowledge engine employs specifically designed goal-fulfillment maps that encode the background knowledge needed to drive the conversations. The conversation engine uses probabilistic finite state machines that model the different types of conversations. Interestingly, Chakrabarti [8, 10] used a form of the Turing test to validate the quality, using Grice's maxims [32], of the dialogue management software. Transcripts of computer generated conversations and human-to-human dialogues in the same domain were judged as roughly (approximately 86%) equivalent, with the computational dialogue significantly ($p < .05$) more focused on the task or goal of the conversation.

To this point in Section 2 we have focused, from an epistemic perspective, on a number of artificial intelligence representational schemes and their related search strategies. A number of critics of traditional AI, including Brooks [5], Dreyfus [22, 23], Merleau-Ponty [52] and Clark [16] suggest that human skilled behavior does not require use of such explicit mental representations and/or focused search. Brooks' robot (seen earlier in this section) finds its way in the world by actively exploring that world. Similarly, Dreyfus [23], taking examples from the acquisition of expertise in playing chess and driving, describes how,

over time, skills become more fluid and spontaneous responses to game and road situations, replacing explicit search algorithms applied to complex static representations.

Merleau-Ponty [52] describes an "intentional arc" where a human does not simply receive input passively from an external world and then processes it. Rather, the active agent is set to respond to the requirements of things, as a world affording, based on past experiences, certain actions. On this viewpoint, the best representation of the world is suggested to be the world itself. Clark [16] makes similar conjectures of an external world that offers "scaffolding" to support intelligent action: The mind moves into the organisms' environment. Clark's example is of a human using paper and pencil or a calculator to do complex arithmetic: an external world supports and enables this intellectual activity.

Stochastic models, and in particular dynamic Bayesian networks (DBN), can offer architectures that address these epistemic concerns of Brooks, Dreyfus et al. In particular, Bayes' theorem supports the active interpretation of environmental cues based of past experiences while the DBN supports acquisition of new information over time. With the analysis of a helicopter's rotor system, research supported by the US Navy, we have the ability to interpret dangerous situations while not supported by any explicit knowledge base of the situation [13]. This is the story of the next section.

3 Modern AI: Probabilistic Models

Although the probabilistic approach to problem solving became popular in the late 1980s and early 1990s, its roots go back to the Reverend Thomas Bayes in the mid-eighteenth century [3]. Early work in probabilistic computer-based problem solving included optical character recognition in the late 1950s and other work at IBM, Bell Laboratories and Carnegie Mellon University. Researchers at Stanford used Bayesian technology in the 1970s in the Prospector program's algorithms for discovering minerals. But it wasn't until the 1990s that probabilistic methods became generally accepted, mostly through the use of statistical methods for language understanding, the research of Judea Pearl [58, 59] and the successes of Bayesian Belief Networks, an important simplification of the complexity issues required of traditional Bayesian technology.

As we present next, the Bayes formula expresses a relationship between newly acquired data (the posterior) and what the problem-solving agent has already seen or experienced (the prior). This algorithmic relationship between the prior and posterior has much intuition supporting it, for example, the ability to understand new language utterances is in a large part due to previous experience, knowledge, and use of that language. Using the terminology of modern philosophy, linguistics, and psychology, we call this learning activity "constructivist".

Constructivism may be described [79] "as the theoretical perspective, central to the work of Jean Piaget [61, 62], that people actively build their per-

ception of the world and interpret actions and events that surround them in terms of what they already know." It follows that a person's present state of knowledge has a major influence on how and what new information is acquired. Unlike empiricism, constructivism is appropriate for producing generalizations (the diameter of a glass is related to its volume) but is more often used to produce most likely responses, given data (glass A holds more water than glass B).

Bayes' theorem [3] offers a plausible model of this constructivist worldview. It is also an important modeling tool for much of modern AI, including AI programs for natural language understanding, robotics, and machine learning. Consider the general form of Bayes' relationship used to determine the probability of a particular hypothesis, h_i, given a set of evidence E:

$$p(h_i \mid E) = \frac{p(E \mid h_i) \cdot p(h_i)}{\sum_{k=1}^{n}(p(E \mid h_k) \cdot p(h_k))}$$

where,
$p(h_i \mid E)$ is the probability that a particular hypothesis, h_i, is true, given evidence E.
$p(h_i)$ is the probability that h_i is true overall, i.e., how likely h_i is to occur.
$p(E \mid h_i)$ is the probability of observing evidence E when hypothesis h_i is true.
n is the number of possible hypotheses.

By Bayes' formula, the probability of an hypothesis being true, given a set of evidence, is equal the probability that the evidence is true given the hypothesis times the probability that the hypothesis occurs. This number is divided by, or normalized by, the probability of the evidence itself. The probability of the evidence occurring is seen as the sum over all hypotheses presenting the evidence times the probability of that hypothesis itself. This "normalization" measure forces the probability $p(h_i \mid E)$ to be a fraction in the 0 to 1 range. As we see shortly, because that denominator, the probability of the evidence, remains the same for all hypotheses, it is often ignored.

As an example, the left side of Bayes' equation gives the probability of a hypothesis h_i given evidence E. Suppose the evidence E for a person is: *headache*, *high temperature*, and *vomiting*, and there are three hypotheses h_i: *a cold*, *viral infection*, or *allergies*, Bayes' formula can then be used to determine which of these hypotheses is most likely, or the best explanation, given evidence E. The right side of Bayes' equation describes how prior knowledge and experience relates to the interpretation of the hypotheses. The expression $p(E \mid h_i)$ asks how often the evidence, a headache, etc, occurs when a particular hypothesis, say allergies, is known to be true. $p(h_i)$ asks how likely that hypothesis itself is to occur.

There are limitations to using Bayes' theorem as just presented as an epistemological characterization of the phenomenon of interpreting new (a posteriori) data in the context of (prior) collected knowledge and experience. Most important is the fact that the epistemic subject is not a calculating machine. We simply don't have all the prior (numerical) values for all the hypotheses and evidence that can fit a problem. In a complex situation such as medical

diagnostics where there can be hundreds of hypothesized diseases and thousands of symptoms, this calculation grows exponentially. We next address this combinatorial issue with three simplifications/extensions of Bayes' rule: naive Bayes, greatest likelihood measures, and Pearl's Bayesian belief networks, including dynamic Bayesian networks.

The calculation of the right hand side of Bayes' formula requires the repeated determination of values for $p(E \mid h_i)$. When the evidence E is a set of parameters (usually represented as a vector), it is often assumed these parameters are independent, given the hypothesis. For example, if the hypothesis is a viral infection v_i, and the evidence set is headaches h, high temperature t, and vomiting v, these three pieces of evidence are assumed to be independent given a viral infection. In most realistic situations this independence assumption is not justified. When this independence is ignored, the algorithm is called naive Bayes. In this example naive Bayes calculates $p(E|h_i)$ as $p(h|v_i) \cdot p(t|v_i) \cdot p(v|v_i)$, a radical improvement both in compute time as well as in the need for obtaining more probability measures such as $p(h \mid v_i \wedge t)$ or $p(h \mid v_i \wedge t \wedge v)$.

A second simplifying approach to using Bayes' rule is to acknowledge that the denominator on the right hand side of the equation, $p(E)$, is the same for all h_i. and thus does not need to be used (or even calculated). This means that, absent the normalization effect of the denominator, the resulting will no longer be a probability measure. Thus, if we wish to determine which of all the hi has the most support given the evidence E, we look for the largest value of $p(E \mid h_i) \cdot p(h_i)$. This is called determining the $argmax$ of all the hypotheses, given the evidence:

$$argmax(h_i) \text{ is the largest for all } h_i \text{ of } p(E \mid h_i) \cdot p(h_i)$$

In a dynamic interpretation, as sets of evidence themselves change across time, we will call this $argmax$ of hypotheses given a set of evidence at a particular time *the greatest likelihood of that hypothesis at that time*. We show this relationship, an extension of the Bayesian *maximum a posteriori* (or MAP) estimate, as a dynamic measure over time t:

$$gl(h_i \mid E_t) = argmax(h_1) p(E_t \mid h_i) \cdot p(h_i)$$

This model is both intuitive and simple: the most likely interpretation of new data, given evidence E at time t, is a function of which interpretation is most likely to produce that evidence at time t and the probability of that interpretation itself occurring. The Bayesian greatest likelihood approach can be viewed as a more sophisticated (and mathematically plausible) alternative to the Stanford certainty-factor algebra commonly used in goal-driven rule-based expert systems [45]. This greatest likelihood relationship can also be interpreted as an example of Piaget's *assimilation*, discussed further in section 4, where newly encountered information fits (is interpreted by) the patterns created from prior experiences.

Pearl [58, 59], proposed the Bayesian Belief Net (BBN) to addresses the complexity of data and inference with full Bayesian reasoning. The BBN makes

two assumptions: first, that reasoning is not circular, that is, that no component of the model can either directly or indirectly influence itself (i.e., that the graph of the model has no cycles), and second, that every variable is independent of all its non-descendents, given knowledge of its parent(s). This graph-based description is intended to capture implicit causality [59]. This assumption captured in the BBN representation the implicit causality of situations. In the example of Figure 7, the occurrence of a traffic accident causes traffic to slowdown and the flashing lights of rescue vehicles.

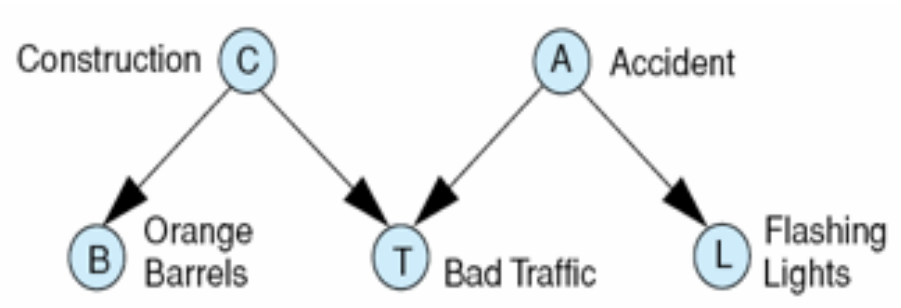
In the BBN example of Figure 7, suppose you are driving in a familiar place where you are aware of the likelihood of traffic slowdowns, construction, and accidents. These likelihoods are reflected in probability tables similar to that of Figure 7, where the top row says that the probability of both construction (C) and bad traffic (T) being $true(t)$ is 0.3. Solving this problem under full Bayesian assumptions would require a 32-row probability table where each of the five variables can be true or false. But factoring by the assumptions of Bayesian belief networks, where C and A, L and C, and A and B are independent of each other, reduces this table to 20 rows, a component of which is presented in Figure 7b.

Now suppose, without any further obvious reasons, you begin to slow down; so bad traffic (T) becomes $true$. This means that in the table of probabilities bad traffic (T) can no longer be false, so the sum of the probabilities for the first and third lines of the table in Figure 5, the construction possibilities (t or f) when there is bad traffic ($T = t$), must be 1.0. This means that with the slowdown of traffic the probability of construction gets much higher (0.75). Similarly the probability of an accident (A) also increases. (These probabilities are not shown).

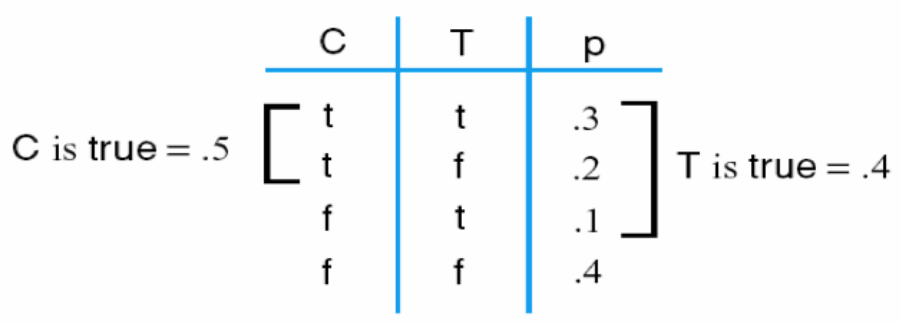
Now suppose you drive along further and you notice Orange Barrels (B) along the road and blocking a lane of traffic (Weighted orange plastic barrels are often used in the U.S. at road projects to control traffic flow). This means that on another probability table (again, not shown here) B is $true(t)$, and in making the probabilities sum to 1.0, the probability of Construction (C) gets much higher. As the probability of Construction gets higher, with the absence of Flashing Lights, the probability of an Accident decreases. The most likely explanation for what you are experiencing now is construction, and the likelihood of an accident goes down, and is said to be *explained away*.

The driving example just described demonstrates what is called a dynamic Bayesian network (DBN). As the perceived information changes over time, first slowing down and then seeing orange traffic control barrels, the probabilities in the table change to reflect each new (posterior) piece of information. Thus at each time period where there is new information, the values reflecting the probabilities of that time period will change. Each state of the table reflects the best explanation for what is currently happening.

As a further example of dynamic Bayesian network (BN) problem solving, Chakrabarti et al. [12, 11] analyze a continuous data stream from a set of distributed sensors, which represents the running "health" of the transmission of a Navy helicopter rotor system through a steady stream of sensor data. This



(a)



(b)

Fig. 7: A Bayesian belief network (BBN) for the bad traffic example and a table giving sample probability values for Construction, C, and Bad Traffic, T. The probabilities for the five parameters of the BBN will be in one 20-row table.

data consists of temperature, vibration, pressure, and other measurements reflecting the state of the various components of the transmission system. An example of this data can be seen in the top portion of Figure 8, where the continuous data stream is broken into discrete and partial time slices.

Chakrabarti et al. [12, 11] then use a Fourier transform to translate these signals into the frequency domain, as shown on the left side of the second row of Figure 8. These frequency readings were compared across time periods to diagnose the running health of the rotor system. The model used to diagnose rotor health is the auto-regressive *hidden Markov model* (AR-HMM) of Figure 9. The observable states of the system are made up of the sequences of the segmented signals in the frequency domain while the hidden states are the imputed health states of the helicopter rotor system itself, as seen in the lower right of Figure 8.

The hidden Markov model (HMM) technology is an important stochastic technique that can be seen as a variant of a dynamic BBN. In the HMM, we attribute values to states of the network that are themselves not directly observable. For example, the HMM technique is widely used in the computer analysis of human speech, trying to determine the most likely word uttered, given a stream of acoustic signals [39]. In the helicopter example, training this

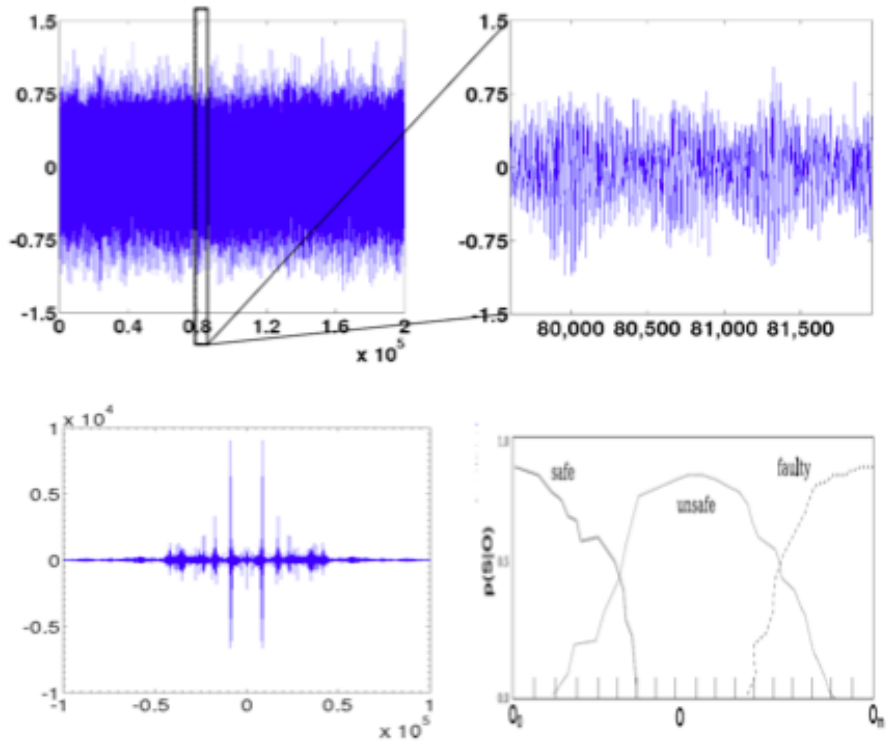


Fig. 8: Real-time data from the transmission system of a helicopters rotor. The top component of the figure presents the original data stream (left) and an enlarged time slice (right). The lower left figure is the result of the Fourier transform of the time slice data (transformed) into the frequency domain. The lower right figure represents the hidden states of the helicopter rotor system.

system on streams of normal transmission data allowed the system to make the correct greatest likelihood measure of failure when these signals change to indicate a possible breakdown. The US Navy supplied data to train the normal running system as well as data sets for transmissions that contained faults. Thus, the hidden state S_t of the AR-HMM reflects the greatest likelihood hypothesis of the state of the rotor system, given the observed evidence O_t at any time t.

To summarize, it is not surprising that the AI problem solving products of the past sixty years have met with limited successes. To give them their due, they have been useful in many of the application domains for which they were intended, designed, and deployed. But as models of human cognition, able to generalize to new related situations, even to generalize and interpret their various results, they were not successful, and, in the context of this paper, could not pass Turing's test. The success of the AI practitioner as the designer and builder of new and useful software languages and artifacts is beyond question;

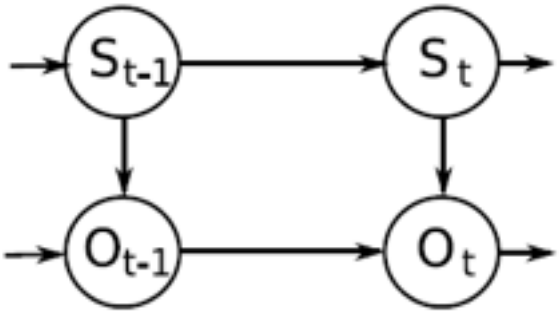


Fig. 9: The data of Figure 8 are processed using an auto-regressive hidden Markov model. States O_t represent the observable values at time t. The S_t states represent the hidden "health" states of the rotor system, $safe$, $unsafe$, $faulty$ at time t.

the notion that this effort emulates the full set of cognitive skills of the human agent is simply naive.

The problem is both epistemological and pragmatic. How does the human agent work within and manipulate elements of a world that is external to, or more simply, is *not*, that agent? And consequently, how can the human agent address the overarching epistemological integration of the agent and its ever-changing environment? And how does (or even *can*) the human agent understand this integration?

4 Towards an Epistemic Stance

Creating any computer program, including the supposed "intelligent program", is the product of human design and computer language skills. Every program built is a product or "artifact" of its human creator. In a programs creation lies an implicit "ontology", or what symbols and patterns of symbols might "mean" when interpreted, as well as an "epistemology", or an a priori commitment to a particular "symbol system mapping to world objects" view of the world. This also represents a specific commitment to a *lebenswelt*, or *lifeworld*, of how symbols and systems of symbols, when interpreted, interact with each other within their environment. If progress in AI is going to evolve, we best acknowledge these differing ontological commitments to what is "real" and in what sense running programs may be said to have "meaning".

An important aspect of a running program is that it can be deconstructed, taken apart, its system of symbols examined, and their relation to a possible world be critiqued. Newell and Simon suggest so much in their Turing Award lecture [57]: "Each new program that is built is an experiment. It poses a question to nature, and its behavior offers clues to an answer. Neither machines nor programs are black boxes; they are artifacts that have been designed, both hardware and software, and we can open them up and look inside. We can

relate their structure to their behavior and we can draw many lessons from a single experiment." Thus, we explore the ontological commitments of the program designer, as well as his/her implicit epistemic stance.

We have found through experimentation that the purer forms of rationalism, although excellent for capturing computation with clear and distinct components of human knowledge, often fail in areas of imprecision and uncertainty. From an empiricist perspective, representing and learning associations has proved a powerful tool, but not always sufficient for discovering appropriate generalizations and capturing higher level relationships. While the pragmatists correctly argue that all action, including language, is "about" intention and goal satisfaction, they have offered few modeling tools or mechanisms for how this might be achieved. Probabilistic reasoning is an implicit acknowledgement that actions in the world are to be understood as a function of the current knowledge and experience of the interpreting agent. We propose, as a natural extension of our discussions, the continued exploration and use of a *constructivist* epistemology as a foundation for building programs that are intended to produce "intelligent" behavior. We also propose that program designers be aware of the full range of tools available for problem solving, many of which have been mentioned in this paper. Especially important are algorithms and representations supporting an active goal-driven and model-refining approach to solutions.

We view a constructivist and model-revising epistemology as a rapprochement between the empiricist, rationalist, and pragmatist viewpoints. The constructivist hypothesizes that all understanding is the result of an interaction between perceived energy patterns from the world and conditioned mental categories imposed on the world by the intelligent agent [61, 62, 27]. Using Piaget's descriptions we *assimilate* external phenomena according to our current understanding and *accommodate* our understanding to phenomena that does not meet our prior expectations.

Constructivists use the term *schemata* to describe the a priori structure used to mediate the experience of the external world. The term schemata is taken from the British psychologist Bartlett [2] and its philosophical roots go back to Kant [40]. On this viewpoint observation is not passive and neutral but active and interpretative.

Perceived information, Kant's *a posteriori* knowledge, never fits precisely into our preconceived and a priori schemata. From this tension the schema-based biases a subject uses to organize experience are either modified or replaced. The use of *accommodation* in the context of unsuccessful interactions with the environment drives a process of cognitive *equilibration*. The constructivist epistemology is one of cognitive evolution and continuous model refinement [28, 29, 69]. An important consequence of constructivism is that the interpretation of any perception-based situation involves the imposition of the observers (biased) concepts and categories on what is perceived. This constitutes an *inductive* bias.

When Piaget proposed a constructivist approach to understanding the external world, he called it a *genetic epistemology*. When encountering new phe-

nomena, the lack of a comfortable fit of current schemata to the world "as it is" creates a cognitive tension. This tension drives a process of schema revision. Schema revision, Piaget's *accommodation*, is the continued evolution of the agent's understanding towards *equilibration*.

Schema revision and continued movement toward equilibration is a genetic predisposition of an agent for an accommodation to the structures of society and the world. It combines both these forces and represents an embodied predisposition for survival. Schema modification is both an *a priori* reflection of our genetics as well as an *a posteriori* function of society and the world. It reflects the embodiment of a survival-driven agent, of a being in space and time.

There is a blending here of the empiricist and rationalist traditions, mediated by the pragmatist requirement of agent intention and survival. As embodied, agents can comprehend nothing except that which first passes through their senses. As accommodating, agents survive through learning the general patterns of an external world. What is perceived is mediated by what is expected; what is expected is influenced by what is perceived: these two functions can only be understood in terms of each other.

Further, we can ask why a constructivist epistemology might be useful in addressing the problem of understanding intelligence itself? How can an agent within an environment understand its own understanding of that situation? We believe that constructivism also addresses this problem of *epistemological access*. For more than a century there has been a struggle in both philosophy and psychology between two factions: the positivist, who proposes to infer mental phenomena from observable physical behavior, and a more phenomenological approach which allows the use of first person reporting to enable access to cognitive phenomena. This factionalism exists because both modes of access to cognitive phenomena require some form of model construction and inference.

In comparison to physical objects like chairs and doors, which often, naively, seem to be directly accessible, the mental states and dispositions of an agent seem to be particularly difficult to characterize. We contend that this dichotomy between direct access to physical phenomena and indirect access to mental phenomena is illusory. The constructivist analysis suggests that no experience of the external (or internal) world is possible without the use of some model or schema for organizing that experience. In scientific enquiry, as well as in our normal human cognitive experiences, this implies that *all* access to phenomena is through exploration, approximation, and continued model refinement.

The dynamic Bayesian network described in Section 3, with implementation details described in Figures 8 and 9, offers an approximation of the constructivist epistemology. The conditioned (prior) variables and relationships in the Bayesian network captures the relational knowledge of the complex system as well as the conditioned aspects of their interactions. Monitoring these systems across time, using an auto-regressive hidden Markov model, captures the changing state of the entire system. Interpreting that state as it evolves over time, based on the conditioning of the running system, supports a goal-

focused interpretation of the state of the helicopter transmission: "everything is going fine", "hey, something might be stressing the system", or "it is time to get this thing onto the ground".

Interestingly, a justification for the use of the hidden Markov modeling technology is that the actual state of complex systems often cannot be deterministically known. We humans only have access to the perceptual cues afforded to our senses by these complex running systems. (This is much like understanding another person's speech - we only have perceptual cues and do not know what is going on within that person). Although the actual state of such systems is "hidden" from the observer, an approximation of that state can be "constructed" through the "assistance" of appropriate hidden Markov model software. A similar constructivist approach for understanding and generating human language dialogs can be seen in Section 2, Figure 5 [8, 10].

There are (at least!) three further issues that need to be addressed in developing a mature epistemological stance. The first is context switching, the second is an agent's active exploration within and between particular probabilistic systems, and the third is the nature of "meaning" or as it is sometimes called, "symbol grounding".

We humans, and this may not be equally necessary for machines, actively participate in our world through what can be called anticipatory contexts. This fact may result from the limited processing capability of our pre-frontal cortex and Broadmann's areas, or through the limited bandwidth of our cortical communication mechanisms. But for whatever reason we focus directly on one thing at a time, with alternative interpretative scenarios remaining in the near subconscious. There needs to be further research, along the lines of Sakhanenko et al. [69, 70] on model failure and context switching in data interpretation: What are the limitations of a particular model? When are models no longer suitable for active interpretation of new data, or Piaget's *accommodation*? What model might afford the next best interpretative context?

The second issue is that we humans, through goal driven activity, explore our environment not just through our already conditioned models for reality, but also by continuously exploring and refining current interpretive contexts. We learn by trying things, by doing things and by mistakes. Gopnik et al. [31, 30, 29] describe how children learn by actively exploring their environment. Other current psychologists and philosophers support and expand this pragmatic and teleological account of human developmental activity [28, 44]. Klein et al. [41] describe how physicists actively explore alternatives in coming to know the current state of their systems (in his examples, particle beam accelerators). Newell and Simon [57] describe how computers, both hardware and software, can be understood through the active study of their behavior. Pearl [59] and others [66] have proposed algorithms for counterfactuals, for actively exploring causality relationships within the contexts of Bayesian networks. The insight here is that intelligent understanding of a constantly changing environment must be active, purposive, and integrative.

Finally, where does "meaning" come from? One intriguing answer might be that many physical systems, once excited, require (need, seek) stasis. This

could also be seen as the satisfaction of the pragmatists' intentionality. AI research, as well as research in physics, [36], has proposed different algorithms for the real-time integration of new (a posteriori) information into previously (a priori) learned patterns of information [19]. Among these algorithms is loopy belief propagation [59] that integrates new data into a system of plausible beliefs, by constantly iterating this system towards equilibrium, or equilibration as Piaget might describe it. The system in stasis includes identification of the most likely explanation (hypothesis) for the new data, given the original state of the system.

To summarize, a system can be in *a priori equilibrium* with its continuing states of learned knowledge. When presented with novel information characterizing a new situation, this a posteriori data perturbs the equilibrium. The loopy belief propagation algorithm then iterates by sending "messages" between near-neighbors' prior and posterior components of the model, until it finds convergence or equilibrium in the form of a particular greatest likelihood hypothesis (meaning?) that explains that novel information.

We conclude with several general comments about human intelligence and the AI research community's implicit assumption of epistemic stances.

5 Conclusion

In concluding, we present three issues relating to the arguments of this paper. First, we conjecture that, at our current level of knowledge, human intelligence and machine intelligence are two different philosophical *kinds*. On this conjecture, although it can be extremely useful to compare both systems' properties and products, the systems still remain different entities and species. In Turing's sense of computational equivalence [18], it has not been shown how to build each system from the other.

Nonetheless, the two systems can share properties: just as birds and airplanes can both fly, so humans and computers can share properties including perception, thinking, and learning. Furthermore, Turing's test for intelligence is agnostic both as to what a computer might be composed of, vacuum tubes, flip-flops, silicon, or even tinker toys, as well as what language processes are used to make it work. All Turing required for "thinking" was that the machine responses be roughly equivalent to those of a human. As noted earlier, it was an impressive insight on Turing's part, that machines, at the primitive stages of the 1940s, would be thought to have the power of thinking.

We humans do not have the freedom to select our own architecture and embodiment for assimilating information about ourselves and our environment. The particulars of our human dispositions and social context mediate our interactions with the world. We possess auditory and visual systems sensitive to a certain bandwidth; we view the world as erect bipeds, having arms, legs, and hands; we are in a world of weather, seasons, sun, and darkness; we are part of a society with evolving goals and purposes; we are individuals that are

born, reproduce, and die. These are our critical support and offer a medium for our understanding, learning, and problem solving.

A machine's embodiment and constraints are quite different. For example, George Miller's [53] "7 plus and minus 2" memory limitation for short term human processing will not constrain a computer; neither will Newell and Simon's observation [56] of human's memory constrained game playing search as "iterative deepening". Nor, conversely, has it been shown that currently understood neural network architectures can be equivalent to human cortical processing. Current machines are quite effective at broad and exhaustive data searches, such as is found in Watson [25], or a search engine's web crawling. And still, when it comes to addressing problems that are exponentially complex such as the games of chess or gomoku, machines have to deal with many of the same constraints we humans must: working through heuristics to find good enough solutions.

Our second concluding comment, an extension of the first, is that since the human and machine share important properties/skills, we can, with a scientifically supported methodology, compare the processes that support these skills. This endeavor began as early as Leibnitz and Hobbes (who conjectured that reasoning was *nothing more than reckoning* [35]) in the 17th century, it became more detailed in the 1940s with the research of McCulloch and Pitts [51] and Hebb's conjectures about human learning [34], and further showed important results in the Information Processing Psychology of the 1950s. It was Allen Newell and Herbert Simon in their 1976 Turing Award lecture [57] that clarified this cognitive information-processing task with their *physical symbol system* hypothesis.

The *physical symbol system* hypothesis proposed that "the necessary and sufficient condition for a physical system to exhibit general intelligent action is that it be a physical symbol system.... Sufficient means that intelligence can be achieved by any appropriately organized physical symbol system.... Necessary means that any agent that exhibits general intelligence must be a physical symbol system".

The *necessary* component of this hypothesis has long been disproved by multiple psychological and computational experiments demonstrating that human intelligent action can be described and explained by non-representation-based models and without the use of explicit symbol manipulation [25, 5, 22, 52]. The *sufficient* component of this hypothesis has lead to a wealth of new experiments and explanations for cognitive tasks. In fact, the *sufficient* component of Newell and Simon physical symbol system hypothesis has supported the discipline of cognitive science and much of the current research in computational linguistics[53, 56, 76, 54, 63].

Finally, our third comment is that most artificial intelligence researchers and software developers are agnostic about the previous two issues. They simply want to make computational artifacts that reflect what a human user might, on observing their effects, call "intelligent". But the primary point of this paper is that in the process of designing and building computational processes that are intended to produce such products, humans do make epistemic

commitments. Whether these are a result of how we think about the world, or whether they are designed towards the strengths of a particular computational system, these programs are human creations and so take on an (often implicit) epistemic stance.

As noted throughout the paper, this stance can support both strengths as well as severe limitations in the final artifact created. It must also be noted that most AI technology is not focused on novel or exciting goals such as playing chess, being a Jeopardy champion, or controlling robots in deep space. Rather, they are focused on and developed to support the quality of human existence, whether this be as a health monitor or advisor, a recommender system for travel routes, an expert system for giving medical or diagnostic advice, or simply supporting better human understanding and communication [10]. As support for human centric interaction, advice, and decision-making the program builder's underlying epistemic assumptions remain important.

References

1. Oxford Dictionary of Philosophy, 2nd edn. Oxford University Press (2014)
2. Bartlett, F.: Remembering. Cambridge University Press (1932)
3. Bayes, T.: Essay towards solving a problem in the doctrine of chances. Philosophic Transactions of the Royal Society of London pp. 370–418 (1763)
4. Blackburn, S.: The Oxford dictionary of philosophy, 15 edn. Oxford University Press. (2008)
5. Brooks, R.A.: A robot that walks: Emergent behaviors from a carefully evolved network. Neural Computation **1**(2), 253–262 (1989)
6. Brooks, R.A.: Intelligence without representation. In: M. Kaufmann (ed.) International Joint Conference on Artificial Intelligence, pp. 596–575 (1991)
7. Buchanan, B.G., Shortliffe, E.H. (eds.): Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison Wesley (1984)
8. Chakrabarti, C.: Artificial conversations for chatter bots using knowledge representation, learning, and pragmatics. Ph.D. thesis, University of New Mexico, Albuquerque, NM (2014)
9. Chakrabarti, C., Luger, G.: An anatomy for artificial conversation generation in the customer service domain. In: 25th Modern Artificial Intelligence and Cognitive Science Conference, pp. 80–85 (2014)
10. Chakrabarti, C., Luger, G.F.: Artificial conversations for customer service chatter bots: Architecture, algorithms, and evaluation metrics. Expert Systems with Applications **42**(20), 6878–6897 (2015)
11. Chakrabarti, C., Pless, D.J., Rammohan, R.R., Luger, G.F.: Diagnosis using a first-order stochastic language that learns. Expert Systems with Applications **32**(3) (2007)
12. Chakrabarti, C., Rammohan, R.R., Luger, G.F.: A first-order stochastic modeling language for diagnosis. In: A. Press (ed.) Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference (2005)
13. Chakrabarti, C., Rammohan, R.R., Luger, G.F.: A first-order stochastic prognostic system for the diagnosis of helicopter rotor systems for the US navy. In: E. Press. (ed.) Second Indian International Conference on Artificial Intelligence. Pune, India (2005)
14. Chomsky, N.: Syntactic Structures. Mouton (1957)
15. Church, A.: An unsolvable problem of elementary number theory. American Journal of Mathematics **58**(2), 345–363 (1936)
16. Clark, A.: Supersizing the mind: Embodiment, action, and cognitive extension. Oxford University Press (2008)
17. Collins, A., Quillian, M.R.: Retreival time from semantic memory. Journal of Verbal Learning and Verbal Behavior **8**, 240–247 (1969)

18. Copeland, B.J.: The Church-Turing Thesis. (2015)
19. Demptster, A.P.: A generalization of bayesian inference. Journal of the Royal Statistical Society **30**(Series B), 1–38 (1968)
20. Descartes, R.: Six Metaphysical Meditations, Wherein it is Proved That there is a God and that Man's Mind is really Distinct from his Body (1680)
21. Dewey, J.: Democracy and Education. Macmillan (1916)
22. Dreyfus, H.: What Computers Still Can't Do. MIT Press (1979)
23. Dreyfus, H.: Intelligence without representation – Merleau-Ponty's critique of mental representation., vol. 1, pp. 367–383 (2002)
24. Epstein, R., Roberts, G., Poland, G. (eds.): Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer. Springer, Dordrecht, Netherlands (2008)
25. Ferrucci, D.: Introduction to "this is watson". IBM Journal Research and Devevelopment **56**(3-4), 1–15 (2012)
26. Fikes, R.E., Nilsson, N.J.: Strips: A new approach to the application of theorem proving to artificial intelligence. Artificial Intelligence **1**(2), 227–232 (1971)
27. von Glaserfeld, E.: An introduction to radical constructivism. The Invented Reality (1978)
28. Glymour, C.: The Mind's Arrows: Bayes Nets and Graphical Causal Models in Psychology. MIT Press (2001)
29. Gopnik, A.: Probabilistic models as theories of children's minds. Behavioral and Brain Sciences **34**(4), 200–201 (2011)
30. Gopnik, A.: A unified account of abstract structure and conceptual change. probabilistic models and early learning mechanisms. commentary on Susan Carey, "the origin of concepts". Behavioral and Brain Sciences **34**(3), 126–129 (2011)
31. Gopnik, A., Glymour, C., Sobel, D.M., Schulz, L.E., Kushnir, T., Danks, D.: A theory of causal learning in children: Causal maps and bayes nets. Psychological Review **111**(1), 3–32 (2004)
32. Grice, P.: Logic and conversation. Syntax and Semantics **3**, 41–58 (1975)
33. Harmon, P., King, D.: Expert systems: Artificial intelligence in business. Wiley (1985)
34. Hebb, D.O.: The Organization of Behavior. Wiley (1949)
35. Hobbes, T.: The Leviathan (1651)
36. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences **79**, 2554–2558 (1984)
37. James, W.: Pragmatism. Hackett (1981)
38. James, W.: The Varieties of Religious Experience. Longmans, Green, and Co (2002)
39. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2 edn. Pearson Prentice Hall (2008)
40. Kant, I.: Immanuel Kant's Critique of Pure Reason. St. Martin's Press (1781)
41. Klein, W.B., Westervelt, R.T., Luger, G.F.: A general purpose intelligent control system for particle accelerators. Journal of Intelligent and Fuzzy Systems (1999)
42. Kolodner, J.L.: Case-based Reasoning. Morgan Kaufmann (1993)
43. Kowalski, R.: Logic for Problem Solving. North-Holland (1979)
44. Kushnir, T., Gopnik, A., Lucas, C., Schulz, L.: Inferring hidden causal structure. Cognitive Science **34**, 148–160 (2010)
45. Luger, G.F.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 6 edn. Addison-Wesley Pearson Education (2009)
46. Masterman, M.: Semantic message detection for machine translation, using interlingua. In: Proceedings of the 1961 International Conference on Machine Translation (1961)
47. McCarthy, J.: Programs with Common Sense,. MIT Press (1968)
48. McCarthy, J.: Circumscription, a form of non-monotonic reasoning. Artificial Intelligence **12**, 27–39 (1980)
49. McCarthy, J.: Applications of circumscription to formalizing common-sense knowledge. Artificial Intelligence **28**, 89–116 (1986)
50. McCarthy, J., Hayes, P.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. The University Press (1969)
51. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics. **5**, 115–133 (1943)

52. Merleau-Ponty, M.: Phenomenology of Perception. Routledge and Kegan Paul (1962)
53. Miller, G.: The magical number seven plus or minus two: Some limits on our capacity for processing information. Psychological Review **63**(2), 81–97 (1956)
54. Miller, G.A.: The cognitive revolution: a historical perspective. In: Trends in Cognitive Sciences., vol. 7, pp. 141–144 (2003)
55. Minsky, M.: The Society of Mind. Simon and Schuster (1986)
56. Newell, A., Simon, H.: Human Problem Solving. Prentice Hall (1972)
57. Newell, A., Simon, H.: Computer science as empirical enquiry: Symbols and search. Communications of the ACM **19**(3), 113–126 (1976)
58. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
59. Pearl, J.: Causality. Cambridge University Press (2000)
60. Peirce, C.S.: Collected Papers 1931 – 1958. Harvard University Press (1958)
61. Piaget, J.: The Construction of Reality in the Child. Basic Books (1954)
62. Piaget, J.: Structuralism. Basic Books (1970)
63. Pinker, S., Bloom, P.: Natural language and natural selection. In: Behavioral and Brain Sciences., vol. 13, pp. 707–784 (1990)
64. Post, E.: Formal reductions of the general combinatorial problem. American Journal of Mathematics **65**, 197–268 (1943)
65. Quillian, M.R.: Word Concepts: A Theory and Simulation of some Basic Semantic Capabilities. Morgan Kaufmann (1967)
66. Rammohan, R.R.: Three algorithms for causal learning. Ph.D. thesis, University of New Mexico, Albuquerque, NM (2010)
67. Rosenbloom, P.S., Lehman, J.F., Laird, J.E.: Overview of soar as a unified theory of cognition. In: Erlbaum (ed.) Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society (1993)
68. Ryle, G.: The Concept of Mind. University of Chicago Press (2002)
69. Sakhanenko, N.A., Luger, G.F., Stern, C.R.: Managing dynamic contexts using failure-driven stochastic models. In: A. Press (ed.) Proceedings of the FLAIRS Conference (2006)
70. Sakhanenko, N.A., Rammohan, R.R., Luger, G.F., Stern, C.R.: A new approach to model-based diagnosis using probabilistic logic. In: A. Press (ed.) Proceedings of the 21st FLAIRS Conference (2008)
71. Schank, R.C., Colby, K.M.: Computer Models of Thought and Language. Freedman (1975)
72. Searle, J.: Speech Acts. Cambridge University Press (1969)
73. Searle, J.: Indirect speech acts, chap. 3, pp. 59–82. Speech Acts. Academic Press, New York (1975)
74. Simon, H.: The Sciences of the Artificial. MIT Press (1981)
75. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley (1984)
76. Sun, R. (ed.): The Cambridge Handbook of Computational Psychology. Cambridge University Press. (2008)
77. Turing, A.: On computable numbers with an application to the entscheidungsproblem. London Mathematical Society **2**(42), 230–265 (1936)
78. Turing, A.: Computing Machinery and Intelligence. Oxford University Press (1950)
79. VandenBos, G.R. (ed.): APA dictionary of psychology, 1st edn. American Psychological Association, Washington, DC (2007)
80. Wilks, Y.: Grammar, Meaning, and the Machine Analysis of Language. Routledge and Kagen Paul (1972)
81. Williams, B.C., Nayak, P.P.: A model-based approach to reactive self-reconfiguring systems. In: M. Press (ed.) Proceedings of the AAAI-96, pp. 971–978 (1996)
82. Williams, B.C., Nayak, P.P.: A reactive planner for a model-based executive. In: M. Press (ed.) Proceedings of IJCAI-97 (1997)