AI | Professor John McCarthy

What is AI? / Basic Questions

Q. What is artificial intelligence?

A. It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Q. Yes, but what is intelligence?

A. Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

Q. Isn't there a solid definition of intelligence that doesn't depend on relating it to human intelligence?

A. Not yet. The problem is that we cannot yet characterize in general what kinds of computational procedures we want to call intelligent. We understand some of the mechanisms of intelligence and not others.

Q. Is intelligence a single thing so that one can ask a yes or no question ``Is this machine intelligent or not?''?

A. No. Intelligence involves mechanisms, and AI research has discovered how to make computers carry out some of them and not others. If doing a task requires only mechanisms that are well understood today, computer programs can give very impressive performances on these tasks. Such programs should be considered ``somewhat intelligent''.

Q. Isn't AI about simulating human intelligence?

A. Sometimes but not always or even usually. On the one hand, we can learn something about how to make machines solve problems by observing other people or just by observing our own methods. On the other hand, most work in AI involves studying the problems the world presents to intelligence rather than studying people or animals. AI researchers are free to use methods that are not observed in people or that involve much more computing than people can do.

Q. What about IQ? Do computer programs have IQs?

A. No. IQ is based on the rates at which intelligence develops in children. It is the ratio of the age at which a child normally makes a certain score to the child's age. The scale is extended to adults in a suitable way. IQ correlates well with various measures of success or failure in life, but making computers that can score high on IQ tests would be weakly correlated with their usefulness. For example, the ability of a child to repeat back a long sequence of digits correlates well with other intellectual abilities, perhaps because it measures how much information the child can compute with at once. However, ``digit span'' is trivial for even extremely limited computers.

However, some of the problems on IQ tests are useful challenges for AI.

Q. What about other comparisons between human and computer intelligence?

Arthur R. Jensen [Jen98], a leading researcher in human intelligence, suggests ``as a heuristic hypothesis'' that all normal humans have the same intellectual mechanisms and that differences in intelligence are related to ``quantitative biochemical and physiological conditions''. I see them as speed, short term memory, and the ability to form accurate and retrievable long term memories.

Whether or not Jensen is right about human intelligence, the situation in AI today is the reverse.

Computer programs have plenty of speed and memory but their abilities correspond to the intellectual mechanisms that program designers understand well enough to put in programs. Some abilities that children normally don't develop till they are teenagers may be in, and some abilities possessed by two year olds are still out. The matter is further complicated by the fact that the cognitive sciences still have not succeeded in determining exactly what the human abilities are. Very likely the organization of the intellectual mechanisms for AI can usefully be different from that in people.

Whenever people do better than computers on some task or computers use a lot of computation to do as well as people, this demonstrates that the program designers lack understanding of the intellectual mechanisms required to do the task efficiently.

Q. When did AI research start?

A. After WWII, a number of people independently started to work on intelligent machines. The English mathematician Alan Turing may have been the first. He gave a lecture on it in 1947. He also may have been the first to decide that AI was best researched by programming computers rather than by building machines. By the late 1950s, there were many researchers on AI, and most of them were basing their work on programming computers.

Q. Does AI aim to put the human mind into the computer?

A. Some researchers say they have that objective, but maybe they are using the phrase metaphorically. The human mind has a lot of peculiarities, and I'm not sure anyone is serious about imitating all of them.

Q. What is the Turing test?

A. Alan Turing's 1950 article *Computing Machinery and Intelligence* [Tur50] discussed conditions for considering a machine to be intelligent. He argued that if the machine could successfully pretend to be human to a knowledgeable observer then you certainly should consider it intelligent. This test would satisfy most people but not all philosophers. The observer could interact with the machine and a human by teletype (to avoid requiring that the machine imitate the appearance or voice of the person), and the human would try to persuade the observer that it was human and the machine would try to fool the observer.

The Turing test is a one-sided test. A machine that passes the test should certainly be considered intelligent, but a machine could still be considered intelligent without knowing enough about humans to imitate a human.

Daniel Dennett's book *Brainchildren* [Den98] has an excellent discussion of the Turing test and the various partial Turing tests that have been implemented, i.e. with restrictions on the observer's knowledge of AI and the subject matter of questioning. It turns out that some people are easily led into believing that a rather dumb program is intelligent.

Q. Does AI aim at human-level intelligence?

A. Yes. The ultimate effort is to make computer programs that can solve problems and achieve goals in the world as well as humans. However, many people involved in particular research areas are much less ambitious.

Q. How far is AI from reaching human-level intelligence? When will it happen?

A. A few people think that human-level intelligence can be achieved by writing large numbers of programs of the kind people are now writing and assembling vast knowledge bases of facts in the languages now used for expressing knowledge.

However, most AI researchers believe that new fundamental ideas are required, and therefore it cannot be predicted when human-level intelligence will be achieved.

Q. Are computers the right kind of machine to be made intelligent?

A. Computers can be programmed to simulate any kind of machine.

Many researchers invented non-computer machines, hoping that they would be intelligent in different ways than the computer programs could be. However, they usually simulate their invented machines on a computer and come to doubt that the new machine is worth building. Because many billions of dollars that have been spent in making computers faster and faster, another kind of machine would have to be very fast to perform better than a program on a computer simulating the machine.

Q. Are computers fast enough to be intelligent?

A. Some people think much faster computers are required as well as new ideas. My own opinion is that the computers of 30 years ago were fast enough if only we knew how to program them. Of course, quite apart from the ambitions of AI researchers, computers will keep getting faster.

Q. What about parallel machines?

A. Machines with many processors are much faster than single processors can be. Parallelism itself presents no advantages, and parallel machines are somewhat awkward to program. When extreme speed is required, it is necessary to face this awkwardness.

Q. What about making a ``child machine'' that could improve by reading and by learning from experience?

A. This idea has been proposed many times, starting in the 1940s. Eventually, it will be made to work. However, AI programs haven't yet reached the level of being able to learn much of what a child learns from physical experience. Nor do present programs understand language well enough to learn much by reading.

Q. Might an AI system be able to bootstrap itself to higher and higher level intelligence by thinking about AI?

A. I think yes, but we aren't yet at a level of AI at which this process can begin.

Q. What about chess?

A. Alexander Kronrod, a Russian AI researcher, said ``Chess is the *Drosophila* of AI.'' He was making an analogy with geneticists' use of that fruit fly to study inheritance. Playing chess requires certain intellectual mechanisms and not others. Chess programs now play at grandmaster level, but they do it with limited intellectual mechanisms compared to those used by a human chess player, substituting large amounts of computation for understanding. Once we understand these mechanisms better, we can build human-level chess programs that do far less computation than do present programs.

Unfortunately, the competitive and commercial aspects of making computers play chess have taken precedence over using chess as a scientific domain. It is as if the geneticists after 1910 had organized fruit fly races and concentrated their efforts on breeding fruit flies that could win these races.

Q. What about *Go*?

A. The Chinese and Japanese game of *Go* is also a board game in which the players take turns moving. *Go* exposes the weakness of our present understanding of the intellectual mechanisms involved in human game playing. *Go* programs are very bad players, in spite of considerable effort (not as much as for chess). The problem seems to be that a position in *Go* has to be divided mentally into a collection of subpositions which are first analyzed separately followed by an analysis of their interaction. Humans use this in chess also, but chess programs consider the position as a whole. Chess programs compensate for the lack of this intellectual mechanism by doing thousands or, in the case of Deep Blue, many millions of times as much computation.

Sooner or later, AI research will overcome this scandalous weakness.

Q. Don't some people say that AI is a bad idea?

A. The philosopher John Searle says that the idea of a non-biological machine being intelligent is incoherent. He proposes the Chinese room argument. The philosopher Hubert Dreyfus says that AI is impossible. The computer scientist Joseph Weizenbaum says the idea is obscene, anti-human and immoral. Various people have said that since artificial intelligence hasn't reached human level by now, it must be impossible. Still other people are disappointed that companies they invested in went bankrupt.

Q. Aren't computability theory and computational complexity the keys to AI? [Note to the layman and beginners in computer

science: These are quite technical branches of mathematical logic and computer science, and the answer to the question has to be somewhat technical.]

A. No. These theories are relevant but don't address the fundamental problems of AI.

In the 1930s mathematical logicians, especially Kurt Gödel and Alan Turing, established that there did not exist algorithms that were guaranteed to solve all problems in certain important mathematical domains. Whether a sentence of first order logic is a theorem is one example, and whether a polynomial equations in several variables has integer solutions is another. Humans solve problems in these domains all the time, and this has been offered as an argument (usually with some decorations) that computers are intrinsically incapable of doing what people do. Roger Penrose claims this. However, people can't guarantee to solve *arbitrary* problems in these domains either. See my Review of *The Emperor's New Mind* by Roger Penrose. More essays and reviews defending AI research are in [McC96a].

In the 1960s computer scientists, especially Steve Cook and Richard Karp developed the theory of NP-complete problem domains. Problems in these domains are solvable, but seem to take time exponential in the size of the problem. Which sentences of propositional calculus are satisfiable is a basic example of an NP-complete problem domain. Humans often solve problems in NP-complete domains in times much shorter than is guaranteed by the general algorithms, but can't solve them quickly in general.

What is important for AI is to have algorithms as capable as people at solving problems. The identification of subdomains for which good algorithms exist is important, but a lot of AI problem solvers are not associated with readily identified subdomains.

The theory of the difficulty of general classes of problems is called *computational complexity*. So far this theory hasn't interacted with AI as much as might have been hoped. Success in problem solving by humans and by AI programs seems to rely on properties of problems and problem solving methods that the neither the complexity researchers nor the AI community have been able to identify precisely.

Algorithmic complexity theory as developed by Solomonoff, Kolmogorov and Chaitin (independently of one another) is also relevant. It defines the complexity of a symbolic object as the length of the shortest program that will generate it. Proving that a candidate program is the shortest or close to the shortest is an unsolvable problem, but representing objects by short programs that generate them should sometimes be illuminating even when you can't prove that the program is the shortest.

What is AI? / Branches of AI

Q. What are the branches of AI?

A. Here's a list, but some branches are surely missing, because no-one has identified them yet. Some of these may be regarded as concepts or topics rather than full branches.

Logical AI

What a program knows about the world in general the facts of the specific situation in which it must act, and its goals are all represented by sentences of some mathematical logical language. The program decides what to do by inferring that certain actions are appropriate for achieving its goals. The first article proposing this was [McC59]. [McC89] is a more recent summary. [McC96b] lists some of the concepts involved in logical aI. [Sha97] is an important text.

Search

AI programs often examine large numbers of possibilities, e.g. moves in a chess game or inferences by a theorem proving program. Discoveries are continually made about how to do this more efficiently in various domains.

Pattern recognition

When a program makes observations of some kind, it is often programmed to compare what it sees with a pattern. For example, a vision program may try to match a pattern of eyes and a nose in a scene in order to find a face. More complex patterns, e.g. in a natural language text, in a chess position, or in the history of some event are also studied. These more complex patterns require quite different methods than do the simple patterns that have been studied the most.

Representation

Facts about the world have to be represented in some way. Usually languages of mathematical logic are used.

Inference

From some facts, others can be inferred. Mathematical logical deduction is adequate for some purposes, but new methods of *non-monotonic* inference have been added to logic since the 1970s. The simplest kind of non-monotonic reasoning is default reasoning in which a conclusion is to be inferred by default, but the conclusion can be withdrawn if there is evidence to the contrary. For example, when we hear of a bird, we man infer that it can fly, but this conclusion can be reversed when we hear that it is a penguin. It is the possibility that a conclusion may have to be withdrawn that constitutes the non-monotonic character of the reasoning. Ordinary logical reasoning is monotonic in that the set of conclusions that can the drawn from a set of premises is a monotonic increasing function of the premises. Circumscription is another form of non-monotonic reasoning.

Common sense knowledge and reasoning

This is the area in which AI is farthest from human-level, in spite of the fact that it has been an active research area since the 1950s. While there has been considerable progress, e.g. in developing systems of *non-monotonic reasoning* and theories of action, yet more new ideas are needed. The Cyc system contains a large but spotty collection of common sense facts.

Learning from experience

Programs do that. The approaches to AI based on *connectionism* and *neural nets* specialize in that. There is also learning of laws expressed in logic. [Mit97] is a comprehensive undergraduate text on machine learning. Programs can only learn what facts or behaviors their formalisms can represent, and unfortunately learning systems are almost all based on very limited abilities to represent information.

Planning

Planning programs start with general facts about the world (especially facts about the effects of actions), facts about the particular situation and a statement of a goal. From these, they generate a strategy for achieving the goal. In the most common cases, the strategy is just a sequence of actions.

Epistemology

This is a study of the kinds of knowledge that are required for solving problems in the world.

Ontology

Ontology is the study of the kinds of things that exist. In AI, the programs and sentences deal with various kinds of objects, and we study what these kinds are and what their basic properties are. Emphasis on ontology begins in the 1990s.

Heuristics

A heuristic is a way of trying to discover something or an idea imbedded in a program. The term is used variously in AI. *Heuristic functions* are used in some approaches to search to measure how far a node in a search tree seems to be from a goal. *Heuristic predicates* that compare two nodes in a search tree to see if one is better than the other, i.e. constitutes an advance toward the goal, may be more useful. [My opinion].

Genetic programming

Genetic programming is a technique for getting programs to solve a task by mating random Lisp programs and selecting fittest in millions of generations. It is being developed by John Koza's group and here's a tutorial:

What is Genetic Programming?

One of the central challenges of computer science is to get a computer to do what needs to be done, without telling it how to do it. Genetic programming addresses this challenge by providing a method for automatically creating a working computer program from a high-level problem statement of the problem. Genetic programming achieves this goal of *automatic programming* (also sometimes called *program synthesis* or *program induction*) by genetically breeding a population of computer programs using the principles of Darwinian natural selection and biologically inspired operations. The operations include reproduction, crossover (sexual recombination), mutation, and architecture-altering operations patterned after gene duplication and gene deletion in nature.

Genetic programming is a domain-independent method that genetically breeds a population of computer programs to solve a problem. Specifically, genetic programming iteratively transforms a population of computer programs into a new generation of programs by applying analogs of naturally occurring genetic operations. The genetic operations include crossover (sexual recombination), mutation, reproduction, gene duplication, and gene deletion.

Preparatory Steps of Genetic Programming

The human user communicates the high-level statement of the problem to the genetic programming system by performing certain well-defined preparatory steps.

The five major **preparatory steps** for the basic version of genetic programming require the human user to specify

(1) the set of terminals (e.g., the independent variables of the problem, zero-argument functions, and random constants) for each branch of the to-be-evolved program,
(2) the set of primitive functions for each branch of the to-be-evolved program,
(3) the fitness measure (for explicitly or implicitly measuring the fitness of individuals in the population),
(4) certain parameters for controlling the run, and
(5) the termination criterion and method for designating the result of the run.

Executional Steps of Genetic Programming

Genetic programming typically starts with a population of randomly generated computer programs composed of the available programmatic ingredients. Genetic programming iteratively transforms a population of computer programs into a new generation of the population by applying analogs of naturally occurring genetic operations. These operations are applied to individual(s) selected from the population. The individuals are probabilistically selected to participate in the genetic operations based on their fitness (as measured by the fitness measure provided by the human user in the third preparatory step). The iterative transformation of the population is executed inside the main generational loop of the run of genetic programming.

The executional steps of genetic programming (that is, the **flowchart of genetic programming**) are as follows:

(1) Randomly create an initial population (generation 0) of individual computer programs composed of the available functions and terminals.

(2) Iteratively perform the following sub-steps (called a *generation*) on the population until the termination criterion is satisfied:

(a) Execute each program in the population and ascertain its fitness (explicitly or implicitly) using the problem's fitness measure.

(b) Select one or two individual program(s) from the population with a probability based on fitness (with reselection allowed) to participate in the genetic operations in (c).

(c) Create new individual program(s) for the population by applying the following genetic operations with specified probabilities:

(i) *Reproduction:* Copy the selected individual program to the new population.

(ii) *Crossover:* Create new offspring program(s) for the new population by recombining randomly chosen parts from two selected programs.

(iii) *Mutation:* Create one new offspring program for the new population by randomly mutating a randomly chosen part of one selected program.

(iv) *Architecture-altering operations:* Choose an architecture-altering operation from the available repertoire of such operations and create one new offspring program for the new population by applying the chosen architecture-altering operation to one selected program.

(3) After the termination criterion is satisfied, the single best program in the population produced during the run (the best-so-far individual) is harvested and designated as the result of the run. If the run is successful, the result may be a solution (or approximate solution) to the problem.

What is AI? / Applications of AI

Q. What are the applications of AI?

A. Here are some.

Game playing

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation--looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second.

Speech recognition

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

Understanding natural language

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

Computer vision

The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

Expert systems

A ``knowledge engineer'' interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. In the present state of AI, this has to be true. The usefulness of current expert systems depends on their users having common sense.

Heuristic classification

One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

What is AI? / More questions

Q. How is AI research done?

A. AI research has both theoretical and experimental sides. The experimental side has both basic and applied aspects.

There are two main lines of research. One is biological, based on the idea that since humans are intelligent, AI should study humans and imitate their psychology or physiology. The other is phenomenal, based on studying and formalizing common sense facts about the world and the problems that the world presents to the achievement of goals. The two approaches interact to some extent, and both should eventually succeed. It is a race, but both racers seem to be walking.

Q. What are the relations between AI and philosophy?

A. AI has many relations with philosophy, especially modern analytic philosophy. Both study mind, and both study common sense.

Q. How are AI and logic programming related?

A. At the very least, logic programming provides useful programming languages (mainly Prolog).

Beyond that, sometimes a theory useful in AI can be expressed as a collection of *Horn clauses*, and goal to be achieved can be expressed as that of finding values of variables satisfying an expression . The problem can sometimes be solved by running the Prolog program consisting of and .

There are two possible obstacles to regarding AI as logic programming. First, Horn theories do not exhaust first order logic. Second, the Prolog program expressing the theory may be extremely inefficient. More elaborate control than just executing the program that expresses the theory is often needed. Map coloring provides examples.

Q. What should I study before or while learning AI?

A. Study mathematics, especially mathematical logic. The more you learn about sciences, e.g. physics or biology, the better. For the biological approaches to AI, study psychology and the physiology of the nervous system. Learn some programming languages--at least C, Lisp and Prolog. It is also a good idea to learn one basic machine language. Jobs are likely to depend on knowing the languages currently in fashion. In the late 1990s, these include C++ and Java.

Q. What is a good textbook on AI?

A. *Artificial Intelligence* by Stuart Russell and Peter Norvig, Prentice Hall is the most commonly used textbbook in 1997. The general views expressed there do not exactly correspond to those of this essay. *Artificial Intelligence: A New Synthesis* by Nils Nilsson, Morgan Kaufman, may be easier to read. Some people prefer *Computational Intelligence* by David Poole, Alan Mackworth and Randy Goebel, Oxford, 1998.

Q. What organizations and publications are concerned with AI?

A. The American Association for Artificial Intelligence (AAAI), the European Coordinating Committee for Artificial Intelligence (ECCAI) and the Society for Artificial Intelligence and Simulation of Behavior (AISB) are scientific societies concerned with AI research. The

Association for Computing Machinery (ACM) has a special interest group on artificial intelligence SIGART.

The International Joint Conference on AI (IJCAI) is the main international conference. The AAAI runs a US National Conference on AI. *Electronic Transactions on Artificial Intelligence*, *Artificial Intelligence*, and *Journal of Artificial Intelligence Research*, and IEEE Transactions on Pattern Analysis and Machine Intelligence are four of the main journals publishing AI research papers. I have not yet found everything that should be in this paragraph.