

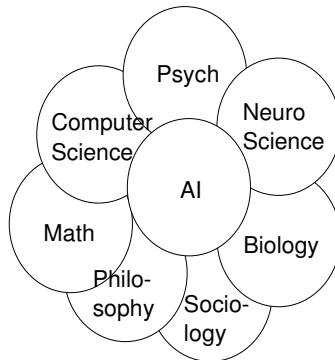
**Introduction to AI**

*What is Artificial Intelligence?*

There are many definitions; some definitions from various textbooks:

<b>Systems that think like humans</b> Machines with minds, in the full and literal sense	<b>Systems that think rationally</b> The study of mental faculties through the use of computational models. The study of the computations that make it possible to perceive, reason, and act.
<b>Systems that act like humans</b> The study of how to make computers do things that, at the moment, people are better. The art of creating machines that perform functions that require intelligence when performed by people.	<b>Systems that act rationally</b> Computational intelligence is the study and design of intelligent agents. Intelligent behavior in artifacts

The book tends along “Systems that act rationally.” From these various definitions, you may guess that AI is a relatively new field, that is still under change. Although it can be traced back to pre-1800’s (for an interesting read, look up "The Turk"), the field was not more formally defined until the late 40’s and into the 60’s and 70’s. AI has many intersections with other disciplines, and many approaches to the AI problem.



We will draw from many different areas that contribute to AI.

**Systems that think like humans**

Most closely related to the field of cognitive science. We need to get inside the actual workings of the human mind and implement this in the computer. One approach is by psychological experiment, the other by introspection. Still another is biologically to reconstruct a computer brain in the same manner as human brains.

## Systems that act humanely

Under this approach the goal is to create a system that acts the same way that humans do, but may be implemented in a totally different way. We'll see the Turing Test shortly which is a way to determine if a system achieves the goal of acting humanely without regard to internal representations. For example, a system might appear to act like a human by inserting random typing errors, but doesn't actually make errors the same way that a human would.

## Systems that think rationally

There is a tradition of using the "laws of thought" that dates back to Socrates and Aristotle. Their study initiated the field of logic. The logicist tradition within AI hopes to build on this approach to create intelligent systems; the main problem has been scaling this approach up beyond toy systems.

## Systems that act rationally

An agent is something that acts. To distinguish an agent from any other program it is intended to perceive its environment, adapt to change, and operate autonomously. A rational agent is one that acts to achieve the best outcome, or best expected outcome when there is uncertainty. Unlike the "laws of thought" approach, these agents might act on incomplete knowledge or to still act when it is not possible to prove what is the correct thing to do. This approach makes it more general than the "laws of thought" approach and more amenable to scientific development than the pure "human-based" approach.

Agent-based activity has focused on the issues of:

- 1) Autonomy. Agents should be independent and communicate with others as necessary.
- 2) Situated. Agents should be sensitive to their own surroundings and context.
- 3) Interactional. Often an interface with not only humans, but also with other agents.
- 4) Structured. Agents cooperate in a structured society.
- 5) Emergent. Collection of agents more powerful than an individual agent.

Another way to think about the field of AI is in term of task domains: Expert tasks (you might hire a professional consultant to do), formal tasks (logic, constraints), mundane tasks (common things you do every day).

Mundane:

Vision, Speech  
Natural Language Processing, Generation, Understanding  
Reasoning  
Motion

Formal:

Board Game-Playing, chess, checkers, gobblet  
Logic  
Calculus  
Algebra  
Verification, Theorem Proving

Expert:

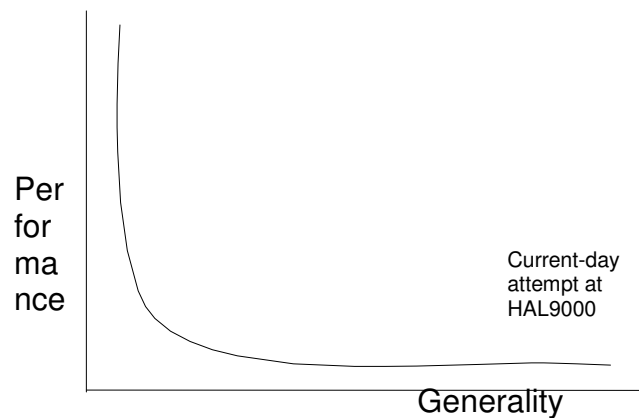
Design, engineering, graphics  
Art, creativity  
Music  
Financial Analysis  
Consulting

People learn the mundane tasks first. The formal and expert tasks are the most difficult to learn. It made sense to focus early AI work on these task areas, in particular, playing chess , performing medical diagnosis, etc. However, it turns out that these expert tasks actually require much less knowledge than do the mundane skills. Consequently, AI is doing very well in the formal and expert tasks; however it is doing very poorly in the mundane tasks.

Example of a mundane task: You are hungry. You have the goal of not being hungry. What do you do to get food? To solve this problem, you have to know what constitutes edible food. You have to know where the food is located. If you do not know where the food is located, you have to find some way to find where it is located, such as looking in phone book or asking someone. You need to navigate to the food. Perhaps the food is in a restaurant. You need to know how to pay for the food, what a restaurant is, what money is, ways to communicate your goals to others, etc... the knowledge necessary to perform this simple task is enormous.

Mundane tasks and the area of broad knowledge understanding are sometimes referred to as “Commonsense Reasoning” and has been termed “AI-Complete” by some researchers.

Generality/Performance curve observed in current AI systems:



Yet another classification of AI is Weak vs. Strong AI. This is essentially the human vs. non-human approach.

- 1) Weak AI. The study and design of machines that perform intelligent tasks. Not concerned with how tasks are performed, mostly concerned with performance and efficiency, such as solutions that are reasonable for NP-Complete problems. E.g., to make a flying machine, use logic and physics, don't mimic a bird.
- 2) Strong AI. The study and design of machines that simulate the human mind to perform intelligent tasks. Borrow many ideas from psychology, neuroscience. Goal is to perform tasks the way a human might do them – which makes sense, since we do have models of human thought and problem solving. Includes psychological ideas in STM, LTM, forgetting, language, genetics, etc. Assumes that the physical symbol hypothesis holds.
- 3) Evolutionary AI. The study and design of machines that simulate simple creatures, and attempt to evolve and have higher level emergent behavior. For example, ants, bees, etc.

### *Philosophical Foundations*

Underlying assumption of Strong AI is the *physical symbol hypothesis*, defined by Newell and Simon in 1976.

Physical symbol hypothesis states: The thinking mind consists of the manipulation of symbols. That is, a physical symbol system has the necessary and sufficient means for general intelligent action.

If this hypothesis is true, then it means that a computer (which merely manipulates symbols) can perform generally intelligent actions. This claim has been rebuked by many researchers citing arguments of consciousness, self-awareness, or quantum theory. David Chalmers has proposed some interesting thought experiments if brain cells were replaced by transistors, and consciousness is graphed vs. transistors.

Turing Test: Proposed by Alan Turing in 1950 as a way to define intelligence. His test is that if the computer should be interrogated by a human through a modem or remote link, and passes the test if the interrogator cannot tell if there is a human or computer at the other end. No computer today can pass the test in a general domain, although computers have used “tricks” to pass in limited domains (e.g. Eliza, Julia). But is something intelligent if it is perceived to be intelligent?

Many complaints about the Turing Test; note that humans often mistake humans on the other end as computers! A famous argument is Searle's Chinese Room. Consider a room, closed off from the world except for an envelope drop. Inside the room is a human with a rule book written in English and stacks of paper for writing. The rule book tells the human how to transcribe from Chinese to English. Naturally, the set of rules is terribly complex, but one can imagine it possible. Now, if someone drops a letter written in Chinese through the slot, the human can follow the rules in the book (perhaps writing intermediate steps) and produce some English output. Question: Does the human understand Chinese? Searle says no, he is just following rules; consequently computers will never “understand” a language like Chinese the same way as humans. (Searle does claim consciousness is an emergent process of neural activity).

Other objections to the Turing test point out that it is biased purely toward symbolic problem-solving skills. Perceptual skill or manual dexterity are left out. Similarly, the test is biased towards humans – it may be possible to have intelligence that is entirely different from human intelligence. After all, why

should a computer be as slow as humans to add numbers? Perhaps one of the largest objections is that of the situational intelligence required. To really pass the Turing Test, some have argued that a machine must be raised and brought up in the same culture and society of humans. How else would a machine know that it is not appropriate to call a “throne” a “chair”? (One answer is to painstakingly enter information like this by hand).

In 1990 Hugh Loebner agreed with The Cambridge Center for Behavioral Studies to underwrite a contest designed to implement the Turing Test. Dr. Loebner pledged a Grand Prize of \$100,000 and a Gold Medal for the first computer whose responses were indistinguishable from a human's. Each year an annual prize of \$2000 and a bronze medal is awarded to the most human computer. The winner of the annual contest is the best entry relative to other entries that year, irrespective of how good it is in an absolute sense. A short snippet of interaction from the winning program in 2009 (the program is “Do-Much-More”):

<http://www.loebner.net/Prize/loebner-prize.html>

**Judge-1:** **Hi, how is it going?**

**Do-Much-More:** **Hello. How are you doing?**

**Judge-1:** **Good, who are you?**

**Do-Much-More:** **Well, I am studying engineering. That ought to give you an idea about who I am.**

**Judge-1:** **No, not really?**

**Do-Much-More:** **I have a terrific yearning to know something more about experiments?**

**Judge-1:** **Where are you from?**

**Do-Much-More:** **I don't know, but don't forget I can't be in three places at once.**

**Judge-1:** **What lang~~~~city do you live in?**

**Do-Much-More:** **What city -- that's rather a tricky question. The trouble is, one city seems to be much like another.**

**Judge-1:** **Not really. Where do you live?**

**Do-Much-More:** **In the old dairy next to the river, I believe.**

**Judge-1:** **Interesting, so you are a farmer then?**

**Do-Much-More:** **Well, if you ask a physiologist what I am, he won't say I'm a farmer. He'll say I consist mostly of water -- enough to fill a large beer tun -- plus some iron, phosphorus, and lots of other prosaic ingredients.**

**Judge-1:** **Wie~~eird answer to a simple question, wouldn't you~ say?**

**Do-Much-More:** **I certainly would. Just watch me!**

As you can see, sometimes the program gives good answers, other times it only picks up on keywords. More information on the Loebner contest is available at <http://www.loebner.net>

### *AI Applications*

Although AI has sometimes been loudly criticized by industry, the media, and academia, there have been many success stories. The criticism has come mainly as a result of hype. For many years, AI was hailed as solving problems such as natural language processing and commonsense reasoning, and it turned out that these problems were more difficult than expected. Here are just a few applications of artificial intelligence.

1. **Game-playing.** IBM's deep-blue has beaten Kasparov, and we have a world-champion caliber Backgammon program. The success here is due to heuristic search and the brute-force power of computers. AI path-finding algorithms and strategy have also been applied to many commercial games, such as WoW or Command & Conquer.
2. **Automated reasoning and theorem-proving.** Newell and Simon are pioneers in this area, when they created the Logic Theorist program in 1963. Logic Theorist proved logical assertions and this helped define propositional calculus and eventually programming languages like Prolog. Formal mathematical logic has been important in fields like chip verification and mission-critical applications such as space missions.
3. **Expert Systems.** An expert system is a computer program with deep knowledge in a specific niche area that provides assistance to a user. Famous examples include DENDREAL, an expert system that inferred the structure of organic molecules from their spectrographic information, and MYCIN, an expert system that diagnosed blood diseases with better accuracy than human experts. More common examples of expert systems include programs like "Turbo Tax" or Microsoft's help system. Typically, a human has to program the expert knowledge into these systems, and they operate only within one domain with little or no learning.
4. **Machine Learning.** Systems that can automatically classify data and learn from new examples has become more popular, especially as the Internet has grown and spawned applications that require personalized agents to learn a user's interests. Some examples include cars capable of driving themselves, face and speech recognition, and Internet portals with pre-classified hierarchies.
5. **Natural Language Understanding, Semantic Modeling.** This area has been successful in limited domains. Most attention has shifted to a shallow understanding of natural language, i.e. witness the various search-engine technologies on the WWW, some that understand rudimentary questions.
6. **Modeling Human Performance.** As described earlier, machine intelligence need not pattern itself after human intelligence. Indeed, many AI programs are engineered to solve useful problems without regard for their similarities to human mental architecture. These systems give us another benchmark to understand and model human performance. Many cognitive scientists use computer techniques to construct their psychological models.
7. **Planning and Robotics.** Planning research began as an effort to design robots that could perform their task. For example, the Sojourner robot on Mars was able to perform some of its own navigation tasks since the time delay to earth makes real-time control impossible. Planning is the task of putting together some sequence of atomic actions to achieve a goal. This area of work extends beyond robots today; for example, consider a web "bot" that puts together a complete travel or vacation package for a customer. It must find reasonable connections and activities in each stop.

8. Languages and Environments. LISP and PROLOG were designed to help support AI, along with constructs such as object-oriented design and knowledge bases. Some of these ideas are now common in mainstream programming languages.
9. Alternative Representations, e.g. Neural Networks and Genetic Algorithms. These are bottom-up approaches to intelligence, based on modeling individual neurons in a brain or the evolutionary process.
10. AI and Philosophy. We have briefly touched on some of the philosophical issues, but there are many more. What happens if we do have intelligent computers? Should they have the same rights as people? What are the ethical issues? What is knowledge? Can knowledge be represented? The questions go on...

#### Abridged History of AI

1943	McCulloch & Pitts: Boolean circuit model of brain
1950	Turing's "Computing Machinery and Intelligence"
1956	Dartmouth meeting: "Artificial Intelligence" adopted
1952—69	Look, Ma, no hands!
1950s	Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
1965	Robinson's complete algorithm for logical reasoning
1966—73	AI discovers computational complexity Neural network research almost disappears
1969—79	Early development of knowledge-based systems
1980--	AI becomes an industry
1986--	Neural networks return to popularity
1987--	AI becomes a science
1995--	The emergence of intelligent agents, genetic algorithms