

CS221 Lecture notes #1

Introduction and history

What is AI? We begin with an almost content-free definition: “AI is the endeavor of building intelligent artifacts or systems.” It’s very hard to make this more precise. People (even within AI) sometimes disagree about precisely what falls within the borderlines of AI.

1 History

AI was born in 1956, at a workshop in Dartmouth organized by John McCarthy. Those gathered agreed to adopt McCarthy’s name for the new field: Artificial Intelligence.

At that point, there was lots of enthusiasm. Things seemed to work out really well. Only a few years before, computers were viewed as large calculators, and now truly intelligent systems seemed within reach. Early programs did amazing things by simply representing knowledge about a domain and searching for a solution. For example, Newell & Simon’s *Logic Theorist* proved qualitative mathematical theorems, and even found a shorter proof for one of the theorems in Russell and Whitehead’s “Principia Mathematica.”

In 1958, McCarthy suggested how the same paradigm could be used for commonsense reasoning: represent knowledge about the everyday world as logical axioms, and use that knowledge to figure out how to act. Amazingly, a general-purpose logical theorem prover was able (for instance) to generate a plan for driving to the airport.

Arguably the first convincing machine learning program, Arthur Samuel’s Checkers playing program started out playing poorly, but learned to play better by playing many games against itself. Growing to play better than Samuel, this program disproved the (still-made) argument that computers can only do what they are told to do.

A particularly good example of how a simple set of rules can produce seemingly complex behavior was Joseph Weizenbaum’s Eliza program, which

simulates a Rogerian psychotherapist. Although Eliza's algorithms are best described as simple pattern matching, and it was not intended as a serious attempt at machine intelligence, it still produced appropriate responses to a variety of statements.

Because of programs like Eliza, there was also a hope of building systems in the near future that would pass the Turing Test for machine intelligence. In the Turing Test, a human judge sits at a computer terminal, and chats via instant messenger with one of two entities: either a human or an AI computer program. The human and computer would each try to convince the human judge that *they* are the human. If the judge is unable to tell whether she is chatting with a human or the AI program, then the AI program passes the Turing Test. (Turing considered this a sufficient, but not necessary, condition for intelligence, since a machine could be intelligent without being able to impersonate a human.)

Things seemed very rosy. Herb Simon, in 1957, said:

It is not my aim to surprise or shock you—but the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in a visible future—the range of problems they can handle will be coextensive with the range to which human mind has been applied.

More precisely: within 10 years a computer would be chess champion, and an important new mathematical theorem would be proved by a computer.

Both of these milestones have now been achieved by computers, but each took closer to 40 years, rather than 10.

After the initial enthusiasm, there was the dawning realization that problems are much harder than one originally thought, and that simple tricks don't work. For instance, one of Eliza's rules was that if the user utters the word "mother," then respond "Tell me more about your family." This sometimes works well, but it can also generate some very unnatural responses. For example, if you say "I wanted to adopt a puppy, but it's too young to be separated from its mother," Eliza may also respond "Tell me more about your family."

Another example was machine translation. Much time and money were spent following Sputnik's launch in 1957 on developing systems to automatically translate Russian documents into English. This turned out to be a very hard problem, since much specialized knowledge seems to be required to understand language. A famous example:

The spirit is willing but the flesh is weak.

was translated into Russian and then retranslated back into English, giving:

The vodka is strong but the meat is rotten.

At that point, people realized two things that made the AI problem much harder than they had originally thought.

1. In order to do a good job in any realistic task, simple syntactic manipulation (i.e., simple rules to shuffle words around or do Russian-to-English dictionary lookups) is not good enough. Instead, we must have enough knowledge about the world to really understand what’s being said, so as to reason more deeply about it. For example, in the translation example, we need to understand that “spirit” refers to the metaphorical or mystical human spirit, rather than to alcohol.
2. Computational intractability. The AI goal was defined before the theory of NP-completeness was developed. At that point, people thought that to deal with larger problems, we need only larger/faster computers. In particular, the phenomena of exponential scaling—in which the computation scales exponentially with the size of the problem—was not yet understood. Many early AI methods required solving NP-hard problems, and therefore did not scale well to larger problems.

2 Approaches to AI

How about the state of AI today? As a field, AI is now significantly more mature, and we have a better understanding of what sort of methods work and might scale well.

There are perhaps two broad approaches to developing AI methods today: (i) Acting like humans, and (ii) Acting rationally.

The Turing Test is perhaps the most famous example of the former: a machine is pronounced intelligent if it is indistinguishable from a human. Today, very little serious work is actually directed specifically at passing the Turing test. There is however a small (but growing) community of researchers that hope to achieve human-level AI by using insights from the humans—specifically, from the human brain. This line of research is inspired by the thesis that much of the human brain may be implementing a learning algorithm. Inspired by this thesis, several research groups are trying to elucidate what the brain’s learning algorithm might be, and implement this algorithm

(or an approximation to it) on a computer, so as to perhaps take a baby step towards solving the problem of building machines that have intelligence comparable to humans. Towards the end of the quarter, we'll talk more about this approach to AI.

Rather than trying to get computers to act like humans, the majority of AI researchers have instead focused on the second approach, of trying to get computers to “act rationally.” This class of methods will be the focus of CS221.

This paradigm has seen numerous successes in terms of building very useful AI systems with significant societal and economic impact. In fact, you probably use AI algorithms dozens of times a day without being aware of it, such through using web search engines, sending US mail or writing checks (where software reads zip codes or handwritten checks automatically), finding driving directions online, receiving Amazon/Netflix/etc. recommendations for books or movies you might like, fraud detection algorithms that check whether your credit card purchases are legitimate, spam filters, and many more.

At a high level, one can view this type of AI as being composed of a set of techniques, such as search, machine learning, constraint satisfaction, and probabilistic models. These techniques are useful for a variety of tasks that are necessary to building various intelligent systems, such as the problems of perception (understanding the physical environment using its sensor inputs), planning, navigation, etc. This is a many-to-many relation: Many very different techniques can be used to perform the same task, and one technique is useful for a wide variety of tasks, and as a component in other techniques.

These techniques can be put together to form some really interesting complete AI systems. In class you saw a video of the Flakey robot carrying out tasks in an office environment. This was an early example of a robot which integrated many kinds of AI tools in order to perform different tasks: speech recognition, computer vision, localization, navigation, and so on. A more recent example is Stanley, Stanford's self-driving car, which won the DARPA Grand Challenge in 2005 (and its successor Junior, which took 2nd place in the 2007 DARPA Urban Challenge). This automated car drove itself about 132 miles across desert terrain, combining techniques from many areas of AI.

In CS221, we'll learn about many of the techniques that went into systems like these.