# A Modern, Agent-Oriented Approach to Introductory Artificial Intelligence

**Stuart Russell*** and **Peter Norvig**†

## Abstract

We describe our experiences in teaching introductory AI and in writing a textbook for the course. The book tries to make the concepts of AI more concrete via two strategies: relating them to the student's existing knowledge, and using examples based on an agent operating in an environment.

## Perceived Problems with Current AI Texts

In the dozen or so times we have taught introductory AI, we have used several of the existing texts, and have always had complaints from students. In a recent student evaluation survey at Berkeley, the text for the AI course was ranked lowest of all texts in computer science courses. Other instructors we have talked to share this sentiment. Some say the current texts are too shallow, or that they present too many ideas without enough motivating examples. The chapters often come across as separate, unrelated subjects, and students don't know what technique to apply to a new problem. Outsiders have criticized AI for concentrating on toy domains, and insiders complain that the texts perpetuate this perception by devoting so much space to Eliza, GPS, and other toy programs of the 1960s. Finally, the texts often have gratuitous differences in terminology and notation that make AI appear alien to the well-rounded computer science student.

In reaction to these problems, we have written a text[1] that we believe presents the field in a much better light. The text has now been used in draft form in over twenty courses and has been well received. In this paper we outline the key pedagogical ideas behind it.

## Unified presentation of the field

Some texts are organized from a historical perspective, describing each of the major problems and solutions that have been uncovered in 40 years of AI research. Although there is value to this perspective, the result is to give the impression of a dozen or so barely related subfields, each with its own techniques and problems. We have chosen to reinterpret some past research and show how it fits within a common framework and how it relates to other work that was historically separate. Often this involves regularizing the notation to emphasize similarities rather than differences. In some cases, we omit work that was important in its day but has since been superseded.

*University of California, Berkeley

†Sun Microsystems Laboratories

[1] *Artificial Intelligence: A Modern Approach*, Prentice Hall.

## Emphasis on Relations to Previous Experience

In all cases, we emphasize the connections that AI has to other areas of computer science, and, to a lesser extent, to mathematics, linguistics, psychology, and philosophy.

For example, many of our students have had a compiler course, and know all about BNF, LR(k) grammars, and various parsing algorithms. A course that teaches ATN grammars fails to connect to these students' previous knowledge. We present parsing by first using BNF notation, and then showing what additions are needed to handle semantics, pragmatics and disambiguation. We recognize that ATNs have an important historical place in AI, but we treat them in a *Bibliographical and Historical Notes* section rather than in the main text. On the other hand, we don't go overboard in stressing similarities. We point out that the purpose of natural language is communication, and that there are many types of speech acts that serve different communicative purposes, in marked contrast to formal languages.

As another example, most of our students have had some training in statistical estimation, such as finding a least-squares polynomial fit to a set of data points. We build on this experience by explaining that all machine learning can be seen as learning the representation of a function. The least-squares method doesn't work well in sparse multi-dimensional domains, so we present a variety of methods that do. All the machine learning methods are compared from this common viewpoint, and we use specific examples to show, for example, that decision trees outperform perceptrons in one domain, but it is the other way around in another domain. In our presentation of neural nets we do examine the analogy between computing units and animal neurons, but the discussion centers around the same questions that we address for other learning methods: what functions can the method represent, and how well can it learn the functions?

## Intelligent agent design

The unifying theme of the book is the concept of an *intelligent agent.* In this view, the problem of AI is to describe and build agents that receive percepts from the environment and perform actions. Each such agent is implemented by a function that maps percepts to actions, and we cover different ways to represent these functions, such as production systems, reactive agents, logical planners and decision-theoretic systems. We show that a reactive agent that maintains no knowledge of the past will perform poorly in certain complex domains. We then introduce the idea of a knowledge-based agent that does maintain some knowledge about the environment, and show how it can perform better. This leads to a discussion of what kind of knowledge is worth maintaining, and how best to represent and reason with it. Only when we have established what knowledge is for (with concrete examples involving a series of agents) do we cover the traditional topics of propositional and first-order logic.

The agent theme forces us to consider some topics that are skipped over in other texts. For example, we have to show where goal-based agents get their goals from. The agent theme also keeps us honest in comparing different AI techniques. Part of the description of every environment is a performance measure by which agents are judged. This gives us a gold standard—to see how well a technique works, we implement an agent that uses it, run the agent in the environment, and see what its average performance score is.

We explain the role of learning as extending the reach of the designer into unknown environments, and show how it constrains agent design, favoring explicit knowledge representation and reasoning.

We treat robotics and vision not as independently defined problems, but as occurring in the service of goal achievement. Throughout, we stress the importance of the task environment characteristics in determining the appropriate agent design. If a few simple reactive rules lead to an agent that performs well in an environment, we won't recommend using a complicated first-order logical problem solver with an ATMS. (But we do show environments for which complex logical reasoning is appropriate.)

There are other possible unifying themes. Figure 1 presents two key dimensions on which definitions of AI vary: (1) whether the field studies thinking or action, and (2) whether it tries to emulate human behavior or strives after ideal behavior. We can imagine very good courses based on different choices, but we feel that (1) studying action is more general than studying thinking, because it derives various cognitive structures from the need to act successfully rather than assuming them *a priori*, and (2) studying rational agents is more appropriate for our audience than studying humans, because many students may have little or no training in biology or psychology.

## Understanding through implementation

The principles of intelligent agent design are clarified by using them to actually build agents. Chapter 2 provides an overview of agent design, including a basic agent and generic environment project. Subsequent chapters include programming exercises that ask the student to consider (and implement) different environments, and to add capabilities to the agent, making it behave more and more interestingly and (we hope) intelligently. Algorithms are presented at three levels of detail: prose descriptions and pseudo-code in the text, and complete Common Lisp programs available by http, anonymous ftp, or on floppy disk. We provide numerous written and programming exercises in each chapter. All the agent programs are interoperable and work in a uniform framework for simulated environments.

We start with a simple environment, the vacuum world, which consists of a rectangular grid of squares, some of which contain dirt. We consider designs for an automated vacuum cleaner that can clean up all the dirt while conserving resources as much as possible. We see what happens as we vary the environmental parameters: How do various agent designs scale with the size of the environment? What happens if the agent's sensors are limited or imperfect? What happens if its actions sometimes fail? What if time is of the essence?

We then move on to slightly more complicated environments, such as the wumpus world, in which agents hunt for gold while avoiding the deadly wumpus and bottomless pits. The wumpus world is ideal for several reasons. First, it is simple enough that the students can understand and implement everything about it. Second, it is complicated enough that it shows the need for logical reasoning by cases (i.e, disjunctive reasoning). When an agent perceives a stench, it means that the wumpus is in a neighboring square, but the exact square is unknown. So the agent has to make several observations and reason about the presence or absence of stenches in different squares before it can conclude where the wumpus is located. Third, students find it engaging. As Prof. Bonnie Webber said about her students at Penn: "They loved it! The whole class participated, asking questions, drawing conclusions for the agent, etc. It was very satisfying." Fourth, the fact that there is a clearly defined performance measure for the agents means that the students can enjoy competing with themselves and their friends to build a top-scoring agent. Finally, whereas we are able to define a wumpus world agent based on logical inference, the environment is complex enough to hint that logic alone does not cover all aspects of the world; we need to augment it with the notions of probability (for the case when we are unable to prove logically which square is safe to move to, but we nevertheless must make a move) and utility (to determine which move will lead to the highest score). We cover these topics in detail later in the book.

## Equal emphasis on theory and practice

Theory and practice are given equal emphasis. All material is grounded in first principles with rigorous theoretical analysis, but the point of the theory is to get the concepts across and explain how they are used in actual, fielded systems. The reader of this book will come away with an appreciation for the basic concepts and mathematical methods of AI, and also with an idea of what can and cannot be done with today's technology, at what cost, and using what techniques.

Our most complex environment is the shopping world: a simulation of a supermarket. For this environment we introduce the idea of a global ontology, covering topics such as categories, measures, parts, change, time, events, actions, substances, and mental objects and events. Thus, we are able to take a concrete problem (an implemented environment), and use it to motivate the most abstract issues in knowledge representation.

## Comprehensive and up-to-date coverage

We cover areas that are sometimes underemphasized, including reasoning under uncertainty, learning, neural networks, natural language, vision, robotics, and philosophical foundations. (The vision and robotics chapters were largely written by Profs. Jitendra Malik and John Canny of U. C. Berkeley.) We cover many of the more recent ideas in the field, for example, simulated annealing, memory-bounded search, global ontologies, dynamic and adaptive probabilistic (Bayesian) networks, fine motion planning for

| | |
|---|---|
| "The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense" (Haugeland 1985)<br><br>"The area of computer science that deals with the ways in which computers can be made to perform cognitive functions ascribed to humans" (Noyes 1992) | "The study of mental faculties through the use of computational models" (Charniak and McDermott 1984)<br><br>"The study of the computations that make it possible to perceive, reason, and act" (Winston 1992) |
| "A field of study concerned with designing and programming machines to accomplish tasks that people accomplish with their intelligence." (Schuster 1987)<br><br>"The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight 1991) | "A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff 1990)<br><br>"The study of intelligent behavior" (Genesereth 1988) |

| | |
|---|---|
| Systems that think like humans. | Systems that think rationally. |
| Systems that act like humans. | Systems that act rationally. |

Figure 1: Some definitions of AI, and the four categories into which they can be arranged.

robots, planning with universal quantification and disjunctive goals, computational learning theory, and reinforcement learning. We also provide extensive notes and references on the historical sources and current literature for the main ideas in each chapter.

## Overview of the book

This book is primarily intended for use in an undergraduate course or course sequence. It can also be used in a graduate-level course (perhaps with the addition of some of the primary sources suggested in the bibliographical notes). Because of its comprehensive coverage and the large number of detailed algorithms, it is useful a primary reference volume for AI graduate students and professionals wishing to branch out beyond their own subfield. We also hope that AI researchers could benefit from thinking about the unifying approach we advocate.

The only prerequisite is familiarity with basic concepts of computer science (algorithms, data structures, complexity) at a sophomore level. Freshman calculus is useful for understanding neural networks in detail. Some experience with nonnumeric programming is desirable, but can be picked up in a few weeks study. We provide implementations of all algorithms in Common Lisp (available over the Internet), but other languages could be used instead.

The book is divided into eight parts. Part I, "Artificial Intelligence," sets the stage for all the others, and offers a view of the AI enterprise based around the idea of intelligent agents—systems that can decide what to do and do it. Part II, "Problem Solving," concentrates on methods for deciding what to do when one needs to think ahead several steps, for example in navigating across country or playing chess. Part III, "Knowledge and Reasoning," discusses ways to represent knowledge about the world—how it works, what it is currently like, what one's actions might do—and how to reason logically with that knowledge. Part IV, "Acting Logically," then discusses how to use these reasoning methods to decide what to do, particularly by constructing *plans*. Part V, "Uncertain Knowledge and Reasoning," is analogous to Parts III and IV, but it concentrates on reasoning and decision-making in the presence of *uncertainty* about the world, as might be faced, for example, by a system for medical diagnosis and treatment.

Together, Parts II to V describe that part of the intelligent agent responsible for reaching decisions. Part VI, "Learning," describes methods for generating the knowledge required by these decision-making components; it also introduces a new kind of component, the *neural network*, and its associated learning procedures. Part VII, "Communicating, Perceiving, and Acting," describes ways in which an intelligent agent can perceive its environment so as to know what is going on, whether by vision, touch, hearing, or understanding language; and ways in which it can turn its plans into real actions, either as robot motion or as natural language utterances. Finally, Part VIII, "Conclusions," analyses the past and future of AI, and provides some light amusement by discussing what AI really is and why it has already succeeded to some degree, and airing the views of those philosophers who believe that AI can never succeed at all.

## Programming Tools

Our main contribution to the stock of instructional AI tools is a testbed for intelligent agents: an architecture for describing environments, and for allowing agent programs to run in the environments. This gives the student a feeling of accomplishment—the agent actually does something—and a way to measure the success of an agent and thus the strength of the technology used to implement the agent.

We also provide simple implementations of some of the standard tools–rule based systems, theorem provers, and so on–and we provide pointers to other inexpensive or free implementations. We provide a host of small utilities to enhance experimentation and understanding, for example, utilities to convert logical sentences into normal form, to print truth tables, and to perform searches. However, the important point is not that we have a lot of programs, but rather that they fit into the overall agent architecture.