WIREs Computational Statistics, Volume 4, I*ssue 2, March/April 2012, pp. 168-180.*

Artificial Intelligence

Gheorghe Tecuci
Learning Agents Center and Computer Science Department
George Mason University, Fairfax, VA 22030

Abstract. Artificial Intelligence is the Science and Engineering domain concerned with the theory and practice of developing systems that exhibit the characteristics we associate with intelligence in human behavior. Starting with a brief history of artificial intelligence, this paper presents a general overview of this broad interdisciplinary field, organized around the main modules of the notional architecture of an intelligent agent (knowledge representation; problem solving and planning; knowledge acquisition and learning; natural language, speech, and vision; action processing and robotics) which highlights both the main areas of artificial intelligence research, development and application, and also their integration.

Artificial Intelligence (AI) is the Science and Engineering domain concerned with the theory and practice of developing systems that exhibit the characteristics we associate with intelligence in human behavior, such as perception, natural language processing, problem solving and planning, learning and adaptation, and acting on the environment. Its main scientific goal is understanding the principles that enable intelligent behavior in humans, animals, and artificial agents. This scientific goal directly supports several engineering goals, such as, developing intelligent agents, formalizing knowledge and mechanizing reasoning in all areas of human endeavor, making working with computers as easy as working with people, and developing human-machine systems that exploit the complementariness of human and automated reasoning.

Artificial Intelligence is a very broad interdisciplinary field which has roots in and intersects with many domains, not only all the computing disciplines, but also mathematics, linguistics, psychology, neuroscience, mechanical engineering, statistics, economics, control theory and cybernetics, philosophy, and many others. It has adopted many concepts and methods from these domains, but it has also contributed back.

While some of the developed systems, such as an expert or a planning system, can be characterized as pure applications of AI, most of the AI systems are developed as components of complex applications to which they add intelligence in various ways, for instance, by enabling them to reason with knowledge, to process natural language, or to learn and adapt.

It has become common to describe an AI system using the agent metaphor[1, pp.34-63]. Fig. 1 shows a notional architecture of an intelligent agent which identifies its main components. In essence, an agent is a knowledge-based system that perceives its environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or other complex environment); reasons to interpret perceptions, draw inferences, solve problems, and determine actions; and acts upon that environment to realize a set of goals or tasks for which it has been designed. Additionally, the agent will continuously improve its knowledge and performance through learning from input data, from a user, from other agents, and/or from its own problem solving experience. While interacting with a human or some other agents, it may not blindly obey commands, but may have the ability to modify requests, ask clarification questions, or even refuse to satisfy certain requests. It can accept high-level requests indicating what the user wants and can decide how to satisfy each request with some degree of independence or autonomy, exhibiting goal-directed behavior and dynamically choosing which actions to take, and in what sequence. It can collaborate with users to improve the accomplishment of their tasks or can carry out such tasks on their behalf, based on knowledge of their goals or desires. It can monitor events or procedures for the users, can advise them on performing various tasks, can train or teach them, or can help them collaborate[2, pp.1-12].

Most of the current AI agents, however, will not have all the components from Fig. 1, or some of the components will have very limited functionality. For example, a user may speak with an automated agent (representing her Internet service provider) that will guide her in troubleshooting her Internet connection. The agent may have advanced speech, natural language, and reasoning capabilities, but no visual or learning capabilities. A natural language interface to a data base may only have natural language processing capabilities, while a face recognition system may only have learning and visual perception capabilities.

Artificial intelligence researchers investigate powerful techniques in their quest for realizing intelligent behavior. But these techniques are pervasive and are no longer considered AI when they reach mainstream use. Examples include time-sharing, symbolic programming languages (e.g., Lisp, Prolog, Scheme), symbolic mathematics systems (e.g., Mathematica), graphical user interfaces, computer games, object-oriented programming, the personal computer, email, hypertext, and even the software agents. While this tends to diminish the merits of AI, the field is continuously producing new results and, due to its current level of maturity and the increased availability of cheap computational power, it is a key technology in many of today's novel applications.

The next section provides a brief history of the evolution of Artificial Intelligence. This is followed by short presentations of its main areas of research which correspond to the agent modules from Fig. 1.

BRIEF HISTORY OF ARTIFICIAL INTELLIGENCE

Artificial intelligence is as old as computer science since from the very beginning computer science researchers were interested in developing intelligent computer systems[3]. The name “artificial intelligence” was proposed by John McCarthy when he and other AI influential figures (Marvin Minsky, Allen Newell, Herbert Simon, a.o .) organized a summer workshop at Dartmouth in 1956.

Early work in artificial intelligence focused on simple “toy” domains and produced some very impressive results. Newell and Simon developed a theorem proving system that was able to demonstrate most of the theorems in Chapter 2 of Russell and Whitehead’s *Principia Mathematica*[1, pp.17-18]. Arthur Samuel developed a checker playing program that was trained by playing against itself, by playing against people, and by following book games. After training, the memory contained roughly 53,000 positions, and the program became "rather better-than-average novice, but definitely not an expert" [4, p.217], demonstrating that significant and measurable learning can result from rote learning alone. Minsky’s students developed systems that demonstrated several types of intelligent behavior for problem solving, vision, natural language understanding, learning and planning, in simplified domains known as "microworlds," such as the one consisting of solid blocks on a tabletop. Robinson[5] developed the resolution method which, theoretically, can prove any theorem in first-order logic.

These successes have generated much enthusiasm and the expectation that AI will soon create machines that think, learn, and create at levels surpassing even human intelligence. However, attempts to apply the developed methods to complex real-world problems have consistently ended in spectacular failures. A famous example is the automatic translation of the phrase "the spirit is willing but the flesh is weak" into Russian, and then back to English, as "the vodka is good but the meat is rotten"[1, p.21]. This has led to an AI winter when previously generous funding for AI research was significantly reduced.

Why have early AI systems failed to scale-up to solve complex real-world problems? One reason is that most of them knew almost nothing about their subject matter. They solved problems by trying all the possible combinations of steps until a solution was found, and were successful because the search space was very small. It was realized that, in order to solve complex-real world problems, a system would need huge amounts of knowledge, as well as heuristics to limit the search for solutions in large problem spaces. It was also realized that building an intelligent agent is very difficult because the cognitive functions to be automated are not understood well-enough. This has led AI researchers to focus on individual cognitive processes, such as learning, and on studying elementary problems in depths, such as concept learning. The consequence was the split of artificial intelligence into many different areas, including knowledge representation, search, game playing, theorem proving, planning, probabilistic reasoning, learning, natural language processing, vision, robotics, neural networks, genetic algorithms, a.o. Each of these areas has established its own research community, with its own conferences and journals, and limited communication with the research communities in other areas. Another split has occurred with respect to the general approach to be used in developing an intelligent system. One is the symbolic approach which relies on the representation of knowledge in symbolic structures and in logic-based reasoning with these structures. The other is the subsymbolic approach that focuses on duplicating the signal-processing and control abilities of simpler animals, using brain-inspired neural networks, biologically-inspired genetic algorithms, or fuzzy logic.

Research in each of these narrower domains facilitated significant progress and produced many successful applications. One of the first successes, which also marks the beginning of the AI industry, was the development and proliferation of expert systems. An expert system incorporates a large amount of domain and human problem solving expertise in a specific area, such as diagnosis, design, planning, or analysis, allowing it to perform a task that would otherwise be performed by a human expert[6-7].

The increasing availability of large data sets, such as the World Wide Web or the genomic sequences, as well as the increased computational power available, has created opportunities for new AI methods that rely more on data than on algorithms. For example, the traditional approach to answering a natural language query from a data repository emphasized deep understanding of the query, which is a very complex problem. But when the repository is as large as the World Wide Web one may simply provide a template for the answer, being very likely that it will be matched by some information on the web.

Progress in various areas of AI has led to a renewed interest in developing agents that integrate multiple cognitive functions. This, in turn, has led to an understanding that various approaches and methods developed in the isolated subfields of AI (natural language processing, knowledge representation, problem solving and planning, machine learning, robotics, computer vision, etc.) need to be interoperable to both facilitate and take advantage of their integration. This has also led to an understanding that the symbolic and subsymbolic approaches to AI are not competing but complementary, and both may be needed in an agent. The result was the development of agent architectures, such as ACT[8], SOAR[9], and Disciple[10], the development of agents for different types of applications (including agents for WWW, search and recommender agents), robots, and multi-agent systems (for instance an intelligent house).

Another aspect of reintegration and interoperability is that algorithms developed in one area are used to improve another area. An example is the use of probabilistic reasoning and machine learning in statistical natural language processing[11].

The next section will briefly review some of the main areas of AI, as identified by the various modules in Fig. 1. The goal is to provide an intuitive understanding of each area, its methods, and its applications.

KNOWLEDGE REPRESENTATION

An intelligent agent has an internal representation of its external environment which allows it to reason about the environment by manipulating the elements of the representation. For each relevant aspect of the environment, such as an object, a relation between objects, a class of objects, a law, or an action, there is an expression in the agent's knowledge base which represents that aspect. For example, Fig. 2 shows one way to represent the situation shown in its upper-right side. The upper part of Fig. 2 is a hierarchical representation of the objects and their relationships (an ontology). Under it is a rule to be used for reasoning about these objects. This mapping between real entities and their representations allows the agent to reason about the environment by manipulating its internal representations and creating new ones. For example, by employing natural deduction and its modus ponens rule, the agent may infer that cup1 is on table 1. The actual algorithm that implements natural deduction is part of the problem solving engine, while the actual reasoning is performed in the Reasoning area (see Fig. 1).

This simple example illustrates an important architectural characteristic of an intelligent agent, the separation between knowledge and control, represented in Fig. 1 by separate modules for the knowledge base and the problem solving engine. While the knowledge base contains the data structures that represent the entities from the environment (as illustrated in Fig. 2), the inference engine implements general methods of solving input problems based on the knowledge from the knowledge base, as will be discussed in the next section**.**

When designing the knowledge representation for an intelligent agent, one has to consider four important characteristics[12]. The first is the *representational adequacy* which characterizes the ability to represent the knowledge needed in a certain application domain. The second is the *inferential adequacy* which denotes the ability to represent the inferential procedures needed to manipulate the representational structures to inferred new knowledge. The third is the *problem solving efficiency* characterizing the ability to represent efficient problem solving procedures. Finally, is the *learning efficiency* characterizing the ability to acquire and learn new knowledge and to integrate it within the agent's knowledge structures, as well as to modify the existing knowledge structures to better represent the application domain.

Since no representation has yet been found that is optimal with respect to all of the above characteristics, several knowledge representation systems have been developed[13-14]. Most of them are based on logic. For example, predicate calculus[15-17] has a high representational and inferential adequacy, but a low problem solving efficiency. The complexity of first order predicate calculus representation makes it very difficult to implement learning methods and they are not efficient. Therefore, most of the existing learning methods are based on restricted forms of first-order logic or even on propositional logic. However, new knowledge can be easily integrated into the existing knowledge due to the modularity of the representation. Thus, the learning efficiency of predicate calculus is moderate.

Production rules[8, 9, 16, 19], which represent knowledge in the form of situation-action pairs, possess similar features. They are particularly well-suited for representing knowledge about what to do in certain situations (e.g., if the car does not start then check the gas), and are used in many agents. However, they are less adequate for representing knowledge about objects.

Semantic networks, frames, and ontologies[14, 20-25] are, to a large extent, complementary to production systems. They are particularly well-suited for representing objects and states, but have difficulty in representing processes. As opposed to production systems, their inferential efficiency is very high because the structure used for representing knowledge is also a guide for the retrieval of knowledge. However, their learning efficiency is low because the knowledge that is added or deleted affects the rest of the knowledge. Therefore, new knowledge has to be carefully integrated into the existing knowledge.

In response to these complementary characteristics, many agents use hybrid representations, such as a combination of ontology and rules, as illustrated in Fig. 2.

Probabilistic representations have been introduced to cope with the uncertainty that derives from a simplified representation of the world, and to enable reasoning with evidence through which many agents experience the world. For example, Fig. 3 shows a Bayesian network due to Judea Pearl. It represents the prior probabilities of specific events (e.g., the prior probability of a burglary at Bob's house is 0.002, and the prior probability of an earthquake is 0.005), and the causal relationships between events (both a burglary and an earthquake cause the house alarm set off with certain probabilities, house alarm set off causes John and Mary to call Bob with certain probabilities). Using the representation in Fig. 3, one may infer, for example, the probability that a burglary has occurred, assuming that both John and Mary called Bob. As in the case of logical representation systems, several probabilistic representation systems have been developed, such as Bayesian[26], Baconian[27, 28], Belief Functions[28], and Fuzzy[30], because none of them can cope with all the characteristic of evidence which is always incomplete, usually inconclusive, frequently ambiguous, commonly dissonant, and has various degrees of believability[31].

Recent research focuses on the more formal representation of the information on the web to facilitate its processing by automated agents, such as the development of the Ontology Web Language[32].

PROBLEM SOLVING AND PLANNING

Artificial intelligence has developed general methods for theorem proving, problem solving and planning, such as, *resolution*, *state space search*, *adversarial search*, *problem reduction*, *constraint satisfaction*, and *case-based reasoning*. One important characteristic of these methods is the use of heuristic information that guides the search for solutions in large problem spaces. While heuristics never guarantee optimal solutions, of even finding a solution, useful heuristics lead to solutions that are good enough most of the time.

In *state space search*, a problem P is represented as an initial state I, a set O of operators (each transforming a state into a successor state), and a set G of goal states. A solution of the problem P is a finite sequence of applications of operators, such as (O_4, O_5, O_1, O_3, O_2), that change the initial state into one of the goal states, as illustrated in Fig. 4. Consider, for example, a robot that can manipulate the objects from Fig. 2. We may ask this robot to bring us the book. The robot needs to find a sequence of actions that transforms the initial state I shown in Fig. 2 into a state G where we have the book in our hands, such as: pick-up cup1, place cup1 on table1, pick-up book1, etc. The definitions of all the actions that the robot can perform (e.g., pick-up, place, etc.), with their applicability conditions and effects on the state of the world, are represented in the knowledge base of the robot. The actual algorithm that applies these operators in order to build the search tree in Fig. 4 is part of the inference engine. The actual tree is built in the Reasoning area (see Fig. 1).

Many algorithms have been developed to solve a problem represented as state space search, including *breath-first search*, *depth first search*, *uniform cost search*, *iterative deepening depth first search*, *greedy best-first search*, *A**, *hill-climbing*, *simulated annealing*, and *genetic algorithms*[1, pp.59-193]. A major difficulty is the size of the search space which makes the use of an exhaustive search unfeasible for real-world problems. The algorithms therefore need to use domain-specific heuristic information that guides them in considering only some of the successors of a node, in a certain order.

This general approach to problem solving has been applied to a wide range of real-world problems, including route finding in computer networks, automated travel advisory systems, airline travel planning systems, planning movements for automatic circuit board drilling, VLSI layout on the chip and channel routing between the cells, robot navigation, and robot assembly.

In *adversarial search*, which is used by the game playing agents, I is the initial board position and the players alternate in selecting the move to make. Before deciding on the next move, the first player projects the game as far as possible into the future. It considers all the possible moves it can make in the initial position (i.e., O_3, O_4, and O_6 in Fig. 4). Each such move (e.g., O_4) would change the game board into a new position (S_4) where it is the turn of the adversary to move. Thus the first player now considers all the possible moves that the adversary can make (i.e., O_5 and O_7 in state S_4), then all its possible responses, and so on. This continues until states are reached which represent end positions in the game (i.e., win, loose, or draw). Then, starting from bottom-up, the first player determines the value (win, draw, or loose) of each intermediate node, based on how the game will end from that node. After all this projection is made, the first player is ready to select, as its first move, the one that leads to the board position having the best result. If both players choose their best moves, the game will end with that result.

This approach, known as *mini-max*, or its more advanced *alpha-beta* version, enables the agent player to select the best move. The problem, however, is that the search space is huge for any non-trivial game. In the case of checker, for instance, it has been estimated that a complete game tree has around 10^{40} nonterminal nodes. If one assumes that these nodes are generated at a rate of 3 billion per second, the generation of the whole tree would still require around $10^{\mathbf{21}}$ centuries![4, p.211]. The search space for chess is much larger, but significantly smaller than the search space for military operations, which involve more players, more possible moves, uncertainty about the state of the world (such as the actual dispositions of the opponent's units), and the use of deception by both forces.

It is therefore clear that an automated agent cannot generate the entire game tree to find the optimal move. What it can do is to build as much of the tree as possible, and use heuristic functions to estimate the values of the generated leaf nodes which do not represent end-game positions. Of course, the closer these nodes are to end positions, the better the estimate produced by the heuristic function.

It is this computational complexity that explains why only in 1997 was an automated agent (Deep Blue of IBM) able to defeat Gary Kasparov, the reigning world champion. The program ran on a very powerful parallel computer generating up to 30 billion positions per move to explore about 14 moves in advance. It contained a database of about 4000 open positions, 700,000 grandmaster games, a large number of end-game solutions, coupled with a heuristic evaluation function based on about 8000 features[33].

In general, game playing agents are better than humans in games where they can search much of the game space (such as Othello). But they are much weaker in games where the search space is very large, such as Go.

Case-based Reasoning is a form of problem solving by analogy in which a new problem is solved by recognizing its similarity to a previously solved problem (which could be classifying the disease of a patient, planning a meal, or designing a circuit), then transferring and adjusting the solution to the new problem[34].

Another general problem solving method that has been employed in expert systems for a wide variety of tasks, including planning, design, critiquing, symbolic integration, and intelligence analysis, is problem reduction[2, 35]. In this approach a problem is solved by successively reducing it top-down to simpler problems, finding the solutions of the simplest problems, and combining these solutions, from bottom-up, to obtain the solution of the initial problem. A simple illustration of this method is shown in Fig. 5 where the problem "Assess whether the United States will be a global leader in wind power within the next decade" is first reduced to three simpler problems (based on a question and its answer), each assessing whether the United States has the reasons, the desire and, respectively, the capability to be a global leader. Each of these problems is further reduced to even simpler problems (guided by other questions and answers). For example, the middle problem is reduced to three other problems. This top-down problem reductions continue until one reaches problems which have known solutions. Then these solutions are successively combined, from bottom-up, to obtain the solutions of the upper-level problems, and of the top-level problem. In the illustration from Fig. 5, these solutions are probabilistic (e.g., "It is almost certain that the people the United States desire the United States to be a global leader in wind power within the next decade.") and are combined using operators such as min, max, average, or weighted sum.

An important characteristic of the problem reduction method is that it shows very clearly the reasoning logic, making it suitable for developing knowledge-based agents that assist can experts and non-experts in problem-solving, and can teach expert problem solving to students.

KNOWLEDGE ACQUISITION AND LEARNING

Much of the power of an intelligent agent derives from the knowledge in its knowledge base (see Fig. 1). A main goal of the knowledge acquisition and machine learning research is precisely to enable an agent to acquire or learn this knowledge from a user, from input data, or from agent's own problem solving experience. This results in improving the competence of the agent in solving a broader class of problems, and in making fewer mistakes in problem solving. It may also result in improving the efficiency of the agent in solving the problems faster and with less memory.

Due to the high complexity of learning, much of the research has focused on the basic task of concept learning, such as learning the concept "cup," or the concept "person who will default on bank loan." In essence, concept learning consists in finding a classification function which distinguishes between the entities that are instances of the concepts from those that are not. Many of the developed learning strategies can be characterized as *empirical inductive learning from examples*, which consists of learning the definition of a concept by comparing positive and negative examples of the concept in terms of their similarities and differences, and inductively creating a generalized description of the similarities of the positive examples[36, 37]. Some methods are based on the information theory to learn the concept in the form of a decision tree[38] that is used to classify the objects. Other methods represent the learned concept as a neural network, whose output unit determines whether the entity at its input units belongs or not to the concept. Learning in a neural network consists in continuously classifying known examples and updating the weights associated with the connection between the units, to improve the recognition accuracy[39, pp.81-127]. Support vector classifiers map the positive and the negative examples of the concept, nonlinearly, into a higher-dimensional feature space via a kernel function, and construct a separating hyperplane there with maximum margin which yields a nonlinear decision boundary in the input space[40]. Bayesian classifiers determine the most likely hypothesis or concept by using the Bayes' rule $P(H|E^*)=P(E^*|H)\bullet P(H)/P(E^*)$ that computes the posterior probability of the hypothesis H based on its prior probability and the observed evidence. This type of learning proved to be very effective in applications where prior probabilities can be computed, and is extensively used for statistical natural language processing[39, pp.154-200].

There are many other learning strategies, besides inductive concept learning from examples. For instance, *explanation-based learning* consists of learning an operational definition of a concept by proving that an example is an instance of the concept and by deductively generalizing the proof[41, 42]. As a result, the agent identifies the important features of the concept, allowing it to recognize much faster the positive examples of the concept, by simply checking that they have these features. *Analogical learning* consists of learning new knowledge about an entity by transferring it from a similar entity, and by testing it[43, 44].

Abductive learning consists of hypothesizing causes based on observed effects[45]. *Conceptual clustering* consists of classifying a set of objects into different classes/concepts and in learning a description of each such class/concept[46]. *Quantitative discovery* consists in discovering a quantitative law relating values of variables characterizing an object or a system[47]. *Reinforcement learning* consists of improving agent's knowledge based on feedback from the environment[48]. *Genetic algorithm-based learning* consists of evolving a population of individuals over a sequence of generations, based on models of heredity and evolution[49].

These learning methods may be used to extend the ontology of an agent with new concepts or facts, or to learn and refine its reasoning rules. Many of these methods are complementary in terms of the input from which the agent learns, the a priori knowledge the agent needs in order to learn, the inferences made during learning, what is actually learned, and the effect of learning on agent's performane. For instance, in the case of empirical inductive learning from examples, in which the primary type of inference is *induction*, the input may consist of *many (positive and/or negative) examples* of some concept C, the knowledge base usually contains only a *small amount of knowledge* related to the input, and the goal is to *learn a description of the concept C* in the form of an inductive generalization of the positive examples which does not cover the negative examples. This description extends or refines the knowledge base and improves the competence of the agent in solving a larger class of problems and in making fewer mistakes.

In the case of explanation-based learning, in which the primary type of inference is *deduction*, the input may consist of only *one example* of a concept C, the knowledge base should contain *complete knowledge about the input*, and the goal is to *learn an operational description of C* in the form of a deductive generalization of the input example. This description is a reorganization of some knowledge pieces from the knowledge base and improves the problem solving efficiency of the agent.

Both analogical learning and abductive learning extend the knowledge base with new pieces of knowledge and usually improve the competence of the agent. In the case of analogical learning, the input may consist of *a new entity I*, the knowledge base should contain an *entity S which is similar to I*, and the goal is to *learn new knowledge about the input I* by transferring it from the known entity S. In abductive learning, the input may be *a fact F*, the knowledge base should contain *knowledge related to the input* and the goal is to *learn a new piece of knowledge* that would explain the input.

Each learning method, used separately, has limited applicability because it requires a special type of input and background knowledge, and it learns a specific type of knowledge. On the other hand, the complementary nature of these requirements and results naturally suggests that by properly integrating these single-strategy methods, one can obtain a synergistic effect in which different strategies mutually support each other and compensate for each other's weaknesses. A large number of multistrategy learning agents that integrate various learning strategies have been developed[50, 51]. Some integrate empirical induction with explanation-based learning, while others integrate symbolic and neural net learning, or deduction with abduction and analogy, or quantitative and qualitative discovery, or symbolic and genetic algorithm-based learning, and so on.

A type of multistrategy learning is that employed by the Disciple agents that can be trained how to perform their tasks, in ways that are similar to how one would train students or apprentices, through specific examples and explanations, and through the supervision and correction of their behavior[2, 52]. For instance, an expert may show a Disciple agent the reasoning tree from Fig. 5. Each question/answer pair following a problem represents the explanation of why that problem is decomposed in the indicated way. But understanding and generalizing these reasoning steps and their explanations, the agent learns general reasoning rules. It then analogically applies these rules to solve similar problems, such as, "Assess whether China will be a global leader in solar power within the next decade." The expert analyses agent's reasoning and characterizes each step as correct or incorrect, also helping the agent in understanding its mistakes. Each of these reasoning steps represents a positive or a negative example for a previously learned rule which is appropriately generalized to cover the positive example or specialized to no longer cover the negative example. As the agent learns new rules and concepts from the expert, their interaction evolves from a teacher-student interaction, toward an interaction where they both collaborate in problem-solving. This process is based on mixed-initiative problem solving[53], where the expert solves the more creative parts of the problem and the agent solves the routine ones and learns from the creative solutions, integrated learning and teaching, where the expert helps the agent to learn (e.g., by providing representative examples, hints, and explanations), and the agent helps the expert to teach it (e.g., by asking relevant questions), and, as already mentioned, multistrategy learning, where the agent integrates complementary strategies, such as learning from examples, learning from explanations, and learning by analogy, to learn general concepts and rules.

NATURAL LANGUAGE, SPEECH, AND VISION

The perceptual processing module in Fig. 1 summarizes agent's capabilities to process natural language, speech, and visual inputs. All are very easy for humans and very difficult for automated agents.

When an agent receives input in natural language, it has to understand it, that is, to build an internal representation of its meaning, which can then be used by the problem solving engine. This process, however, is very difficult for several reasons. Natural language is ambiguous at all levels: morphology, syntax, semantics, and discourse[54, 55]. Just by hearing a common word such as "run" we cannot say whether it is a noun or a verb. The WordNet semantic dictionary[56] gives 16 senses for the noun interpretation and 41 senses for the verb interpretation. What does the word "diamond" mean? Does it mean the mineral consisting of nearly pure carbon in crystalline form? Does it mean a gem or other piece cut from this mineral? Does it mean a lozenge-shaped plane figure (♦)? Does it mean the playing field in Baseball? Therefore the meaning of a word needs to be interpreted in the context of its surrounding words. But sentences themselves may be ambiguous. What does "Visiting relatives can be boring" mean? Does it mean that the act of visiting relatives can be boring? Maybe it means that the relatives who visit us can be boring. Consider also the possible meanings of the following sentence: "She told the man that she hated to run alone." Therefore the meanings of individual sentences need themselves to be interpreted in the context of the paragraphs that contain them. This is also necessary because of additional complexities of natural language, such as, the use of paraphrases (where the same meaning may be expressed by different sentences), ellipses (the use of sentences that appear ill-formed because they are incomplete, which requires the extraction of the missing parts from previous sentences), and references (where entities are referred by pronouns, such as “it” or “they,” without giving their names). But even considering larger paragraphs may not be enough for understanding their meaning, unless the agent has a large amount of knowledge about the domain of discourse.

As illustrated in Fig. 7[57], understanding a natural language sentence (such as “Tarzan kissed Jane.”) involves a sequence of stages, including part-of-speech disambiguation, parsing (which analyzes the syntactic structure of the input sentence and produces a parse tree that identifies its syntactic components, such as “noun phrase” and “verb phrase”), semantic interpretation (which produces an internal representation of the meaning of the sentence), and contextual/world knowledge interpretation (where the internal representation is extended with world knowledge from the knowledge base, such as “Tarzan loves Jane,” to produce a more complete and accurate understanding of the meaning of the sentence). The resulting structure represents agent’s understanding of the meaning of the input sentence which may further be used to answer a question, translate the sentence into a different language, or understand a story.

The above stages are based on various types of grammars which specify the rules for well-formed expressions, and are augmented with semantic interpretations. Early natural language understanding systems were based on manually-defined grammars and were limited in their coverage of a given natural language. Therefore successful systems were developed for restricted areas of discourse, such as airline reservation or question-answering in a specific domain.

The availability of large language corpora on the world-wide web has had a very significant impact on the field of natural language processing, both in terms of the methods and techniques involved (which are now mainly based on probability and statistics), and in terms of its applications. In these approaches, prior probabilities are associated with specific words, and with the rules of the grammar. This allows one to determine the probability distribution of various meanings of an ambiguous word or sentence, and to determine the most likely meaning in a given context[11, 58].

A type of very successful application of statistical natural language processing and concept learning is classifying a text into one of several categories. Examples of such applications include identifying the language of a text, identifying whether a product review is positive or negative, or whether an email message is spam or non-spam, with recognition accuracy in the range of 98%-99% and, in some cases, exceeding 99.9%[1, p. 866].

In information retrieval, a user specifies a query in natural language and the agent has to return the most relevant documents from a given knowledge repository, such as the World Wide Web. In question answering, the user desires an answer to its question, rather than a document. Because of the huge amount of information on the web, which is likely to contain many answers to a question, and in many forms, an agent has to understand what the question is about (topic), and what the user is interested in (focus). It can then provide a simple template for the expected answer rather than trying all the possible paraphrases based on deep natural language understanding. This is because it is very likely that the simple answer template will match some text on the web.

As mentioned several times, an intelligent agent needs a huge amount of knowledge in its knowledge base, which is very difficult to define. With the progress made in statistical natural language processing, some of this knowledge, such as parts of the ontology (i.e., concepts, instances, relationships – see Fig. 2) can be automatically learned.

Finally, another important area of natural language processing is automatic translation from one natural language into another. All translation systems use some models of the source and target languages. Classical approaches attempt to understand the source language text, translate it into an interlingua representation, and then generate sentences in the target language from that representation. In statistical approaches, translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual parallel text corpora.

Speech recognition consists in identifying the spoken words from their acoustic signals, and is difficult because these signals are both ambiguous and noisy. Therefore the most successful approaches are also based on statistical methods. Both machine translation and speech recognition are among the biggest successes of artificial intelligence, and are part of many applications[59].

Research on vision concerns the development of algorithms allowing an agent to extract information from its environment to recognize and manipulate objects, and to navigate[60]. Many algorithms have been developed that detect the edges, texture, and surfaces of objects, and segment an image into its main components. However, recognizing component objects in a scene, or understanding the scene, remain very difficult problems.

ACTION PROCESSING AND ROBOTICS

The action processing module in Fig. 1 corresponds to the agent's actions upon that environment aimed at realizing the goals or tasks for which it was designed. Such an action could be the generation of an answer to a question, the solution of an input problem, the manipulation of an object, or the navigation to a new position. Since most of these actions have already been addressed in the above sections, here we only address object manipulation and navigation, which are the main concern of the robotics area[61].

One may distinguish between three main types of robots: (1) manipulators which are robotic arms attached to their workspace, such as those used in car assembly and painting; (2) mobile robots with wheels, legs, or wings, used to move objects around, such as the unmanned air vehicles for surveillance, crop-spraying, or military operations, or the autonomous underwater vehicles; and (3) mobile manipulators that combine mobility with manipulation to accomplish more complex tasks. A challenging problem in robotics is localization and mapping, which consists in finding out where things are and building a map of the environment. Another challenging problem is path planning from one point in space to another point, which may involve compliant motion, where the robot moves while maintaining physical contact with an object (e.g., an obstacle, a box it pushes, or a screw it inserts).

Robots have many applications in industry (e.g., for part assembly or painting), agriculture (e.g., as special machines), transportation (e.g., autonomous vehicles), health care (e.g., as devices for surgery), hazardous environments (e.g., for clearing minefields or cleaning up nuclear waste), space exploration, and entertainment. They can provide personal services (e.g., vacuum cleaning), or can act as human augmentation devices (e.g., by providing additional force to facilitate walking or arm movement).

CONCLUSION

The main goal of Artificial Intelligence is to develop computational agents that exhibit the characteristics we associate with intelligence in human behavior. Such an agent has an internal representation of its external environment which is at their basis of its reasoning abilities. In general, an agent solves complex real-world problems by using large amounts of knowledge and heuristic methods. It is highly desirable that the agent's knowledge and reasoning are understandable to humans, and the agent is able to explain its behavior, what decisions it is making, and why. The agent may reason with data items that are more or less in contradiction with one another, and may provide some solution without having all the relevant data. The agent should be able to communicate with its users, ideally in natural language, and it may continuously learn.

Why are intelligent agents important? Because humans have limitations that agents may alleviate, such as limited attention span, ability to analyze only a small number of alternatives at a time, or memory for the details that is not affected by stress, fatigue or time constraints. Humans are slow, sloppy, forgetful, implicit, and subjective. But they have common sense and intuition, and may find creative solutions in new situations. By contrast, agents are fast, rigorous, precise, explicit, and objective. But they lack common sense and the ability to deal with novel situations[62, 63]. Humans and agents may thus engage in mixed-initiative reasoning that takes advantage of their complementary strengths and reasoning styles. As such, intelligent agents enable us to do our tasks better, and help us in coping with the increasing challenges of globalization and the rapid evolution toward the knowledge economies[64].

REFERENCES

1. Russell SJ, Norvig P. Artificial Intelligence: A Modern Approach, Prentice-Hall, 2010.
2. Tecuci G. Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies, Academic Press, San Diego, 1998.
3. Turing A. Computing machinery and intelligence. Mind 1950, 59:433-460.
4. Samuel AL. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development 1959, 3:210-229.
5. Robinson JA. A machine-oriented logic based on the resolution principle. JACM 1965, 12:23-41.
6. Buchanan BG, Sutherland GL, Feigenbaum EA, Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. In Meltzer B, Michie D, Swan M, ed. Machine Intelligence 1969, 4:209-254.
7. Buchanan BG, Shortliffe EH. Rule Based Expert Systems. Reading, MA: Addison-Wesley, 1984.
8. Anderson JR. The Architecture of Cognition. Harvard University Press, 1983.
9. Laird J, Newell A, Rosenbloom PS. SOAR: An architecture for general intelligence. Artificial Intelligence Journal 1987, 33:1-64.

10. Tecuci G. DISCIPLE: A Theory, Methodology and System for Learning Expert Knowledge. Thèse de Docteur en Science. University of Paris-South, 1988.

11. Manning C, Schutze H. Foundations of Statistical Natural Language Processing. MIT Press, 1999.

12. Rich E, Knight K. Artificial Intelligence. McGraw-Hill, 1993.

13. Brachman RJ, Levesque HJ., ed. Readings in Knowledge Representation. Morgan Kaufmann, San Mateo, CA, 1985.

14. Brachman RJ, Levesque H. Knowledge Representation and Reasoning. Morgan Kaufman, San Francisco, CA, 2004.

15. McCarthy J. Programs with common sense. In Minsky ML, ed. Semantic Information Processing. MIT Press, Cambridge, MA, 1968 403-418.

16. Kowalski R. Logic for Problem Solving. Elsevier, North-Holland, Amsterdam, London, New York, 1979.

17. Genesereth MR, Nilsson NJ. Logical Foundations of Artificial Intelligence. Morgan Kaufmann, San Mateo, CA, 1987.

18. Waterman D, Hayes-Roth F. Pattern-Directed Inference Systems. New York, Academic Press, 1978.

19. Brownston L, Farrell R, Kant E, Martin N. Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming. Reading, Addison-Wesley, MA, 1985.

20. Quillian MR. Semantic memory. In Minsky M, ed. Semantic Information Processing. Cambridge, Mass, MIT Press, 1968 227-270.

21. Minsky M. A framework for representing knowledge. In Winston PH, ed. The Psychology of Computer Vision. McGraw-Hill, New York, 1975 211-277.

22. Bobrow DG, Winograd T. An overview of KRL, a knowledge representation language. Cognitive Science 1977, 1:3-46.

23. Sowa JF. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove, CA, 1984.

24. Brachman RJ, Schmolze JG. An overview of the KL-ONE knowledge representation system. Cognitive Science 1985, 9:171-216.

25. Lenat DB, Guha RV. Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Addison-Wesley, Reading, Massachusetts, 1990.

26. Pearl J. Causality: Models, Reasoning, and Inference. Cambridge University Press, New York, 2009.

27. Cohen LJ. The Probable and the Provable. Clarendon Press, Oxford, 1977.

28. Cohen LJ. An Introduction to the Philosophy of Induction and Probability. Clarendon Press, Oxford, 1989.

29. Shafer G. A Mathematical Theory of Evidence. Princeton University Press, Princeton, NJ, 1976.

30. Zadeh L. The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems. Fuzzy Sets and Systems 1983, 11:199 - 227.

31. Schum DA. The Evidential Foundations of Probabilistic Reasoning. Northwestern University Press, 1994, 2001.

32. OWL 2 Web Ontology Language, http://www.w3.org/TR/owl2-overview/ (accessed August 14 2011)

33. Campbell MS, Hoane AJ, Hsu F-H. Deep Blue. Artificial Intelligence Journal 2002, 134:57-83.

34. Leake DB. Case-Based Reasoning: Experiences, Lessons and Future Directions. MIT Press, 1996.

35. Nilsson NJ. Problem Solving Methods in Artificial Intelligence. McGraw-Hill, New York, NY, 1971.

36. Mitchell TM. Version Spaces: An Approach to Concept Learning. Doctoral Dissertation. Stanford University, 1978.

37. Michalski RS. A Theory and Methodology of Inductive Learning. In Michalski RS, Carbonell JG, Mitchell TM, ed. Machine Learning: An Artificial Intelligence Approach, Vol. 1. Tioga Publishing Co, 1983, 83-129.

38. Quinlan JR. Induction of decision trees. Machine Learning 1986, 1:81–106.

39. Mitchell TM. Machine Learning. McGraw-Hill, 1997.

40. Hearst MA, Dumais ST, Osman E, Platt J, Scholkopf B. Support vector machines. IEEE Intelligent Systems 1998, 13:18-28.

41. Mitchell TM, Keller RM, Kedar-Cabelli ST. Explanation-based generalization: a unifying view. Machine Learning 1986, 1:47-80.

42. DeJong G, Mooney R. Explanation–based learning: an alternative view. Machine Learning 1986, 1:145-176.

43. Winston PH. Learning and reasoning by analogy. Communications of the ACM 1980, 23:689-703.

44. Gentner D. Structure mapping: a theoretical framework for analogy. Cognitive Science 1983, 7:155-170.

45. Josephson J, Josephson S. Abductive Inference. Cambridge University Press, 1994.

46. Fisher DH. Knowledge acquisition via incremental conceptual clustering. Machine Learning 1987, 2:139-172.

47. Langley P, Simon HA, Bradshow GL, Zytkow JM. Scientific Discovery: Computational Explorations of the Creative Processes. MIT Press, Cambridge, MA, 1987.

48. Kaelbling LP, Littman ML, Moore AV. Reinforcement learning: a survey. Journal of AI Research 1996, 4:237-285. Online journal at http://www.cs.washington.edu/research/jair/-home.html

49. DeJong K. Evolutionary Computation: Theory and Practice. MIT Press, 2006.

50. Tecuci G. Plausible justification trees: a framework for the deep and dynamic integration of learning strategies. Machine Learning Journal 1993, 11: 237-261.

51. Michalski RS., Tecuci G., ed. Machine Learning: A Multistrategy Approach. Morgan Kaufmann Publishers, San Mateo, CA, 1994.

52. Tecuci G, Boicu M, Boicu C, Marcu D, Stanescu B, Barbulescu M. The Disciple-RKF learning and reasoning agent. Computational Intelligence 2005, 21:462-479.

53. Tecuci G, Boicu M, Cox MT. Seven aspects of mixed-initiative reasoning: an introduction to the special issue on mixed-initiative assistants. AI Magazine 2007, 28:11-18.

54. Tufiş D, Ion R, Ide N. Fine-grained word sense disambiguation based on parallel corpora, word alignment, word clustering and aligned wordnets. In Proceedings of the 20th International Conference on Computational Linguistics, COLING2004, Geneva, 2004 1312-1318.

55. Tufiş D. Algorithms and data design issues for basic nlp tools. In Nirenburg S, ed. Language Engineering for Lesser-Studied Languages, IOS Press, NATO Science for Peace and Security Series - D: Information and Communication Security 2009, 21:3-50.

56. http://wordnetweb.princeton.edu/perl/webwn (accessed October 1 2011)

57. Luger GF, Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 6th ed, Pearson Education , Inc, 2009.

58. Jurafsky D, Martin JH. Speech and Language Processing. Prentice Hall, 2008.

59. Huang XD, Acero A, Hon H. Spoken Language Processing. Prentice Hall, 2001.

60. Forsyth D, Ponce J. Computer Vision: A Modern Approach. Prentice Hall, 2002.

61. Bekey G. Robotics: State of the Art and Future Challenges. Imperial College Press, 2008.

62. Turoff M. Design of Interactive Systems. In Emergency Management Information Systems Tutorial. The Hawaii International Conference on System Sciences, HICSS-40. Hawaii, 2007.

63. Phillips-Wren G, Jain LC, Nakamatsu K, Howlett RJ., ed. Advances in Intelligent Decision Technologies, SIST 4. Springer-Verlag, Berlin Heidelberg, 2010.

64. Toward Knowledge Societies, http://unesdoc.unesco.org/images/0014/001418/141843e.pdf (accessed October 1 2011)

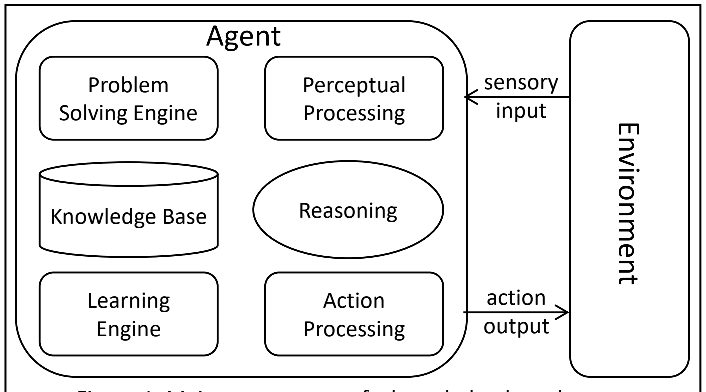


Figure 1. Main components of a knowledge-based agent.

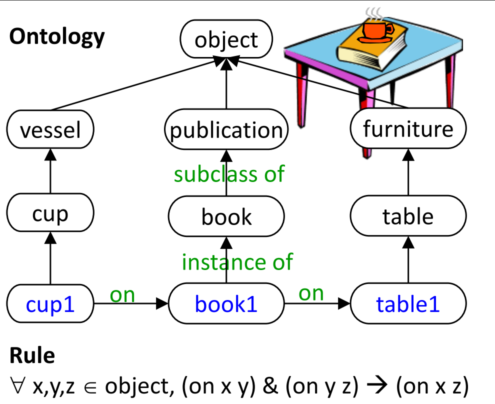


Figure 2. A situation and its representation.

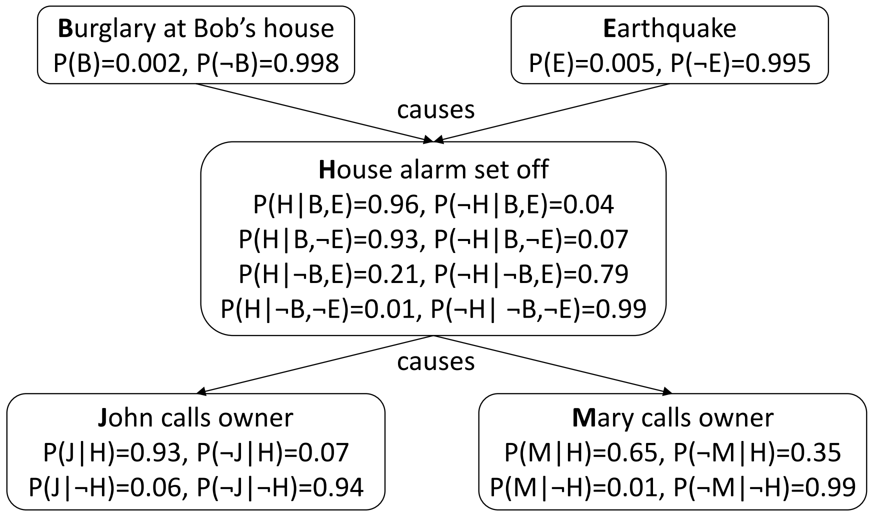

Burglary at Bob's house
P(B)=0.002, P(¬B)=0.998
Earthquake
P(E)=0.005, P(¬E)=0.995
causes
House alarm set off
P(H|B,E)=0.96, P(¬H|B,E)=0.04
P(H|B,¬E)=0.93, P(¬H|B,¬E)=0.07
P(H|¬B,E)=0.21, P(¬H|¬B,E)=0.79
P(H|¬B,¬E)=0.01, P(¬H| ¬B,¬E)=0.99
causes
John calls owner
P(J|H)=0.93, P(¬J|H)=0.07
P(J|¬H)=0.06, P(¬J|¬H)=0.94
Mary calls owner
P(M|H)=0.65, P(¬M|H)=0.35
P(M|¬H)=0.01, P(¬M|¬H)=0.99

Figure 3. Bayesian network showing the causal relationships between events and their prior probabilities.

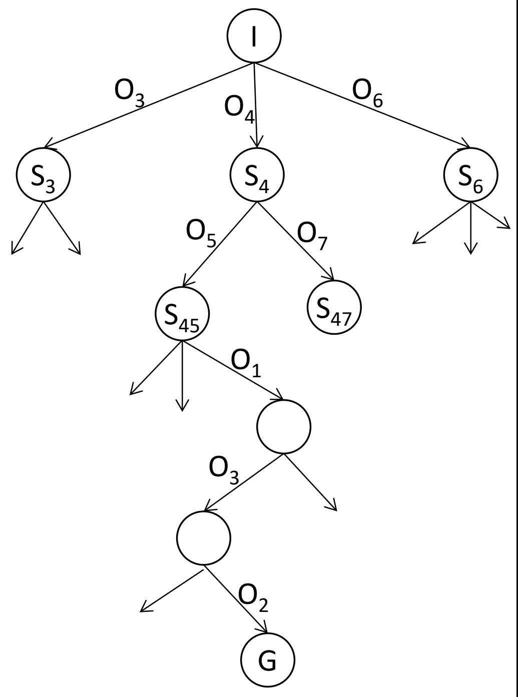
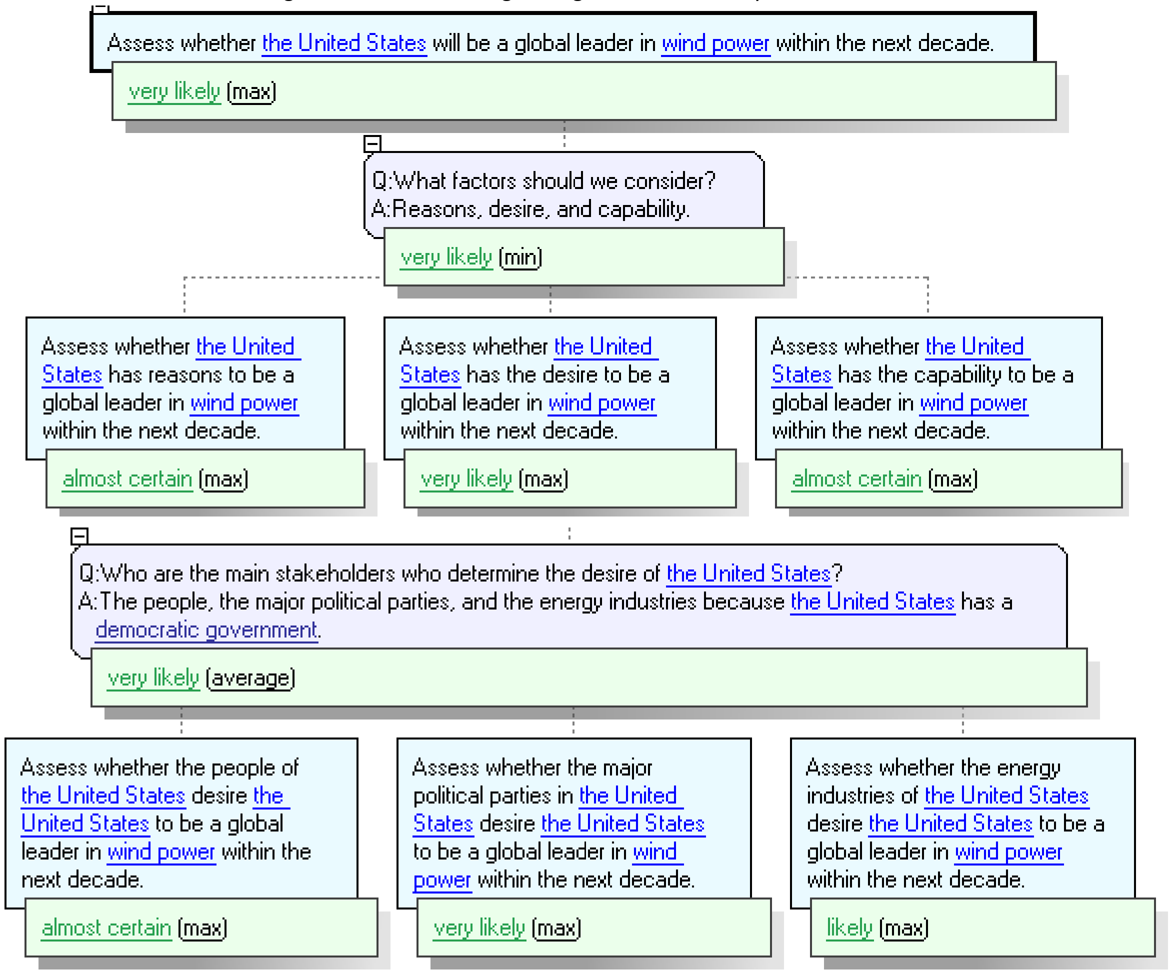


Figure 4. Problem solving as search.

Figure 5. Problem solving through reduction and synthesis.


Assess whether the United States will be a global leader in wind power within the next decade.
very likely (max)
Q:What factors should we consider?
A:Reasons, desire, and capability.
very likely (min)
Assess whether the United States has reasons to be a global leader in wind power within the next decade.
almost certain (max)
Assess whether the United States has the desire to be a global leader in wind power within the next decade.
very likely (max)
Assess whether the United States has the capability to be a global leader in wind power within the next decade.
almost certain (max)
Q:Who are the main stakeholders who determine the desire of the United States?
A:The people, the major political parties, and the energy industries because the United States has a democratic government.
very likely (average)
Assess whether the people of the United States desire the United States to be a global leader in wind power within the next decade.
almost certain (max)
Assess whether the major political parties in the United States desire the United States to be a global leader in wind power within the next decade.
very likely (max)
Assess whether the energy industries of the United States desire the United States to be a global leader in wind power within the next decade.
likely (max)

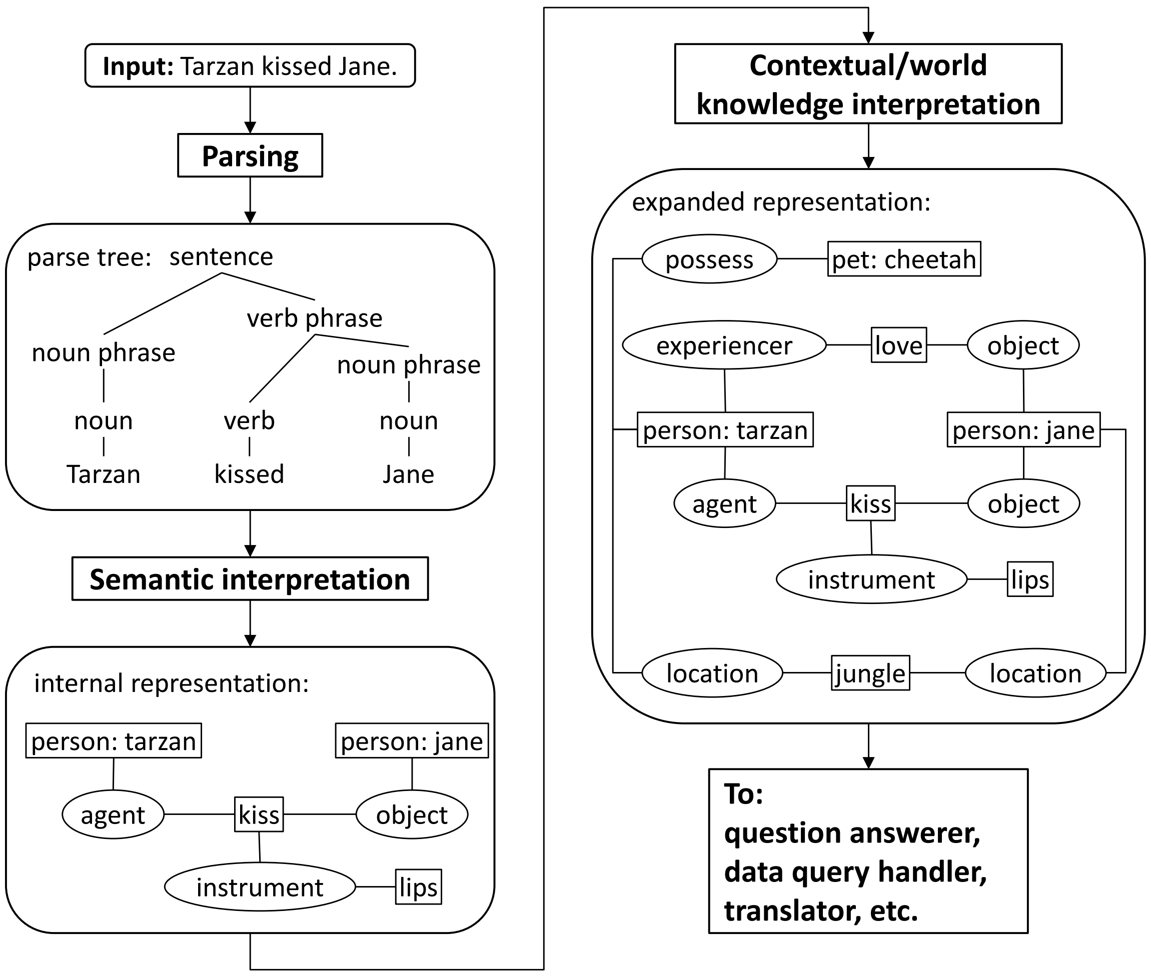


Figure 7. Stages in understanding a natural language sentence (from Luger FG, Artificial Intelligence, 2009: 624).