CONCEPT LEARNING

by

R. S. Michalski

In AI Encyclopedia, John Wiley & Sons, New York, pp. 185-194. January, 1986.

ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE

VOLUME 1

Stuart C. Shapiro, *Editor-in-Chief*
David Eckroth, *Managing editor*
George A. Vallasi, *Chernow Editorial Services, Developmental Editor*

Wiley-Interscience Publication
John Wiley & Sons
New York / Chichester / Brisbane / Toronto / Singapore

However, Henry was in a school that was set up to favor a very different pattern of development. Children were encouraged to act as experts and advisors to the other children whenever they had special knowledge. The computers were located out in the open rather than in computer labs or in classrooms where quiet was imposed. This made it much easier to see what other children were doing and to interact with anyone doing intriguing work. Thus, it was not the computer as such but the computer culture of the school that drew Henry into a situation where he was in demand. So this young man who had always been afraid of pursuing contacts with other children found himself being pursued.

Finally, a more subtle example is drawn from the author's work with Logo. From the outset this language was designed to encourage communication between users. Logo programs are modular so they can be borrowed and shared. Logo is also designed to make it as easy as possible to talk about how you made your program work—what the bugs were, what the difficulties were, and how you solved them. Thus, the content of actual computer work, even on what might seem like a very technical level such as designing a computer language, is a factor that can make for greater socialization or greater isolation.

In all these conceptual issues one needs to remember one thing. Any question such as "What effect will the computer have upon this or that?" is a badly posed question. It is not the computer. In each case it is not what the computer will do to one, it is what one will do with the computer.

BIBLIOGRAPHY

1. *The New York Times*, Section 12, Education, Sunday, April 14, 1985.
2. S. Turkle, *The Second Self: Computers and the Human Spirit*, Simon and Schuster, New York, pp. 129–136, 1984.

General References

C. Daiute, *Writing and Computers*, Addison-Wesley, Reading, MA, 1985.

R. Lawler, *Computer Experience and Cognitive Development: A Child's Learning in a Computer Culture*, Ellis Horwood Ltd., distributed by John Wiley & Sons, New York, 1985.

T. O'Shea, *Learning and Teaching with Computers*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, New York, 1980.

D. Peterson, *The Intelligent Schoolhouse*, Reston Publishing, Reston, VA, 1984.

S. Weir, *Cultivating Minds: Another* [illegible] Harper & Row, NY, 1986.

S. P[illegible]
Mass[illegible] stitute of Technology

CONCEPT LEARNING

What is Concept Learning?

Among the fundamental characteristics of intelligent behavior are the abilities to pursue goals and to plan future actions. To exhibit these characteristics, an intelligent system—human or machine—must be able to classify some objects, behaviors, or events as equivalent for achieving given goals and some others as differing. For example, to satisfy hunger, an animal must be able to classify some objects as edible despite the great variety of their forms and the changes they undergo in the environment. Thus, an intelligent system must be able to form concepts, that is, classes of entities united by some principle. Such a principle might be a common use or goal, the same role in a structure forming a theory about something, or just similar preceptual characteristics. In order to use the concepts, the system must also develop efficient methods for recognizing concept membership of any given entity. The question then is how concepts and concept recognition methods are learned.

The study and computer modeling of processes by which an intelligent system acquires, refines, and differentiates concepts is the subject matter of concept learning. Concept learning is a subdomain of machine learning (qv). The research in this area originated with studies of concept development in humans (e.g., Refs. 1–3). It subsequently continued in the context of both AI efforts to build machines with concept-learning capabilities and cognitive science studies to construct computational models of learning. Selected publications covering this development are listed in Refs. 4–23.

At present, concept learning is one of the central research topics in machine learning, a subarea of AI concerned with the development of computational theories of learning and the building of learning machines (see Machine learning). In research on concept learning, the term "concept" is usually viewed in a more narrow sense than outlined above, namely, as an equivalence class of entities, such that it can be comprehensibly described by no more than a small set of statements. This description must be sufficient for distinguishing this concept from other concepts. Individual entities in the class are called instances of the concept.

The assumption that a concept is an equivalence class implies that its every instance is equally representative of the concept and that the concept description has precise boundaries, that is, it either matches or does not match any given entity. (This notion is more general than the classical definition, which postulates that a concept is characterized by singly necessary and jointly sufficient conditions and thus excludes a disjunctive description.) Such an idealization greatly facilitates research on concept learning, as it defines the learning task simply as the acquisition of a formal structure describing an equivalence class. It is, however, only a very rough approximation that ignores many important aspects of the human notion of a concept (24). At the conclusion of this entry the weaknesses of this definition are briefly addressed, and ideas are pointed out that attempt to capture the notion of a concept more adequately.

Within research on concept learning two major orientations can be distinguished: cognitive modeling and the engineering approach. They parallel the orientations of efforts in cognitive science and AI, respectively. Cognitive modeling strives to develop computational theories of concept learning in humans or animals. It blends original cognitive psychology techniques with efforts to develop well-defined computational methods and computer programs embodying those methods. In contrast, the engineering approach attempts to explore and experiment with all possible learning mechanisms, irrespective of their occurrence in living organisms.

Concept Learning Can Be Classified by Type of Inference Performed

In any learning process the student applies the knowledge possessed to information obtained from a source, for example, a teacher, in order to derive new useful knowledge. This new knowledge is then stored for subsequent use. Learning a new concept can proceed in a number of ways, reflecting the type of inference the student performs on the information supplied. For example, one may learn the concept of a butterfly by being given a description of it, by generalizing examples of specific butterflies, by constructing this concept in the process of observing and analyzing different types of insects, or by yet another way. The type of inference performed by the student on the information supplied defines the strategy of concept learning and constitutes a useful criterion for classifying learning processes.

Several basic concept-learning strategies have been identified in the course of machine-learning research. These are presented below in the order of increasing complexity of inference as performed by the learner. In some general sense, this order reflects the increasing difficulty for the student to learn the concept and the decreasing difficulty for the instructor to teach the concept. In any practical act of learning, more than one strategy is often simultaneously employed. It should also be noted that this classification of strategies applies not only to learning of concepts but also to any act of acquiring knowledge.

Direct Implanting of Knowledge. This is an extreme case in which the learner does not have to perform any inference on the information provided. The knowledge supplied by the source is directly accepted by the learner. This strategy, also called rote learning, includes learning by direct memorization of given concept descriptions and learning by being programmed or constructed. For example, this strategy is employed when a specific algorithm for recognizing a concept is programmed into a computer or a database of facts about the concept is built. In Samuel's CHECKERS program (5) rote learning was employed to save the results of previous game tree searches in order to deepen and speed up subsequent searches.

Learning by Instruction (or Learning by Being Told). Here the learner acquires concepts from a teacher or other organized source, such as a publication or textbook, but does not directly copy into memory the information supplied. The learning process may involve selecting the most relevant facts and/or transforming the source information to more useful forms. The system NANOKLAUS (25), which builds a hierarchical knowledge base by conversing with a user, is an example of machine learning employing this strategy.

Learning by Deduction. The learner acquires a concept by deducing it from the knowledge given and or possessed. In other words, this strategy includes any process in which knowledge learned is a result of a truth-preserving transformation of the knowledge given, including performing computation. A very simple example of this strategy determining that the factorial of 6 is 720 by executing an already known algorithm and having this fact for future use. This technique is called "memo functions" (26). A form by deduction is explanation-based learning which transforms an abstract, not directly usable, concept definition to an operational definition using a concept example for guidance (27). In general, deductive learning is performing a sequence of deductions or computations on the information given and/or stored in background knowledge, and memorizing the result.

More advanced deductive learning is exemplified by analytic or explanation-based learning methods (e.g., 27). These methods start with the abstract concept definition and domain knowledge, and by deduction derive an operational concept definition. A concept example is used to guide the deductive process. For instance, knowing that a cup is an open, stable and liftable vessel, an explanation-based method can produce an "operational" description of a cup. Such a description characterizes the cup in terms of lower level, more measurable features, such as the presence of concavity, of a handle and a flat bottom. Current research attempts to combine such analytical learning with inductive learning in order to learn concepts when the domain knowledge is incomplete, intractable or inconsistent.

Learning by Analogy. The learner acquires a new concept by modifying the definition of a known similar concept. That is, rather than formulating a rule for a new concept from scratch, the student adapts an existing rule by modifying it appropriately to serve the new role. For example, if one knows the concept of an orange, learning the concept of a tangerine can be accomplished easily by just noting the similarities and distinctions between the two. Another example is learning about electric circuits by drawing analogies from pipes conducting water.

Learning by analogy can be viewed as inductive and deductive learning combined and for this reason is placed between the two. Through inductive inference (see below) one determines general characteristics or transformations unifying concepts being compared. Then, by deductive inference, one derives from these characteristics features expected of the concept being learned. Winston (18) describes a method for learning concepts by analogy based on matching semantic networks. Learning by analogy plays an important role in problem solving (e.g., Ref. 22).

Learning by Induction. In this strategy the learner acquires a concept by drawing inductive inferences from supplied facts or observations. Depending on what is provided and what is known to a learner, two different forms of this strategy can be distinguished: learning from examples and learning from observation and discovery.

Learning from Examples. The learner induces a concept description by generalizing from teacher- or environment-provided examples and (optionally) counterexamples of the concept. It is assumed that the concept already exists; it is known to the teacher or there is some effective procedure for testing the concept membership. The task for the learner is to determine a general concept description by analyzing individual concept examples.

An example of this strategy takes place when a senior doctor examines medical records and makes interviews with patients in the presence of one or more interns, noting that "this is a patient with hepatitis"; "this is another patient with hepatitis, but notice that . . . ", and so on. The latter part of this entry briefly discusses a few methods for learning from examples.

Learning by Observation and Discovery. In this strategy the learner analyzes given and or observed entities and determines that some subsets of these entities can be grouped use-

fully into certain classes (i.e., concepts). Because there is no teacher who knows the concepts beforehand, this strategy is also called unsupervised learning. Once a concept is formed, it is given a name. Concepts so created can then be used as terms in subsequent learning of other concepts.

An important form of this strategy is clustering (i.e., partitioning a collection of objects into classes) and the related process of constructing classifications. Classifications are typically organized into hierarchies of concepts. Such hierarchies exhibit an important property of inheritance. If an object is recognized as a member of some class, the properties associated specifically with this class, as well as with classes at the higher level of hierarchy, are (tentatively) assigned to the given object. For example, if one learns that Freddy is an elephant, then, without seeing Freddy, one will typically assume that Freddy has four legs, a trunk, and all the distinguishing properties of elephants, vertebrates, and generally, animals. Hierarchical classifications vary in height: Some may be tall, like the classification of living organisms, and some more flat, like the social hierarchy. The topics of clustering (in particular, conceptual clustering) and classification construction are treated in a separate entry in the encyclopedia (see Clustering).

Another form of learning by observation and discovery is descriptive generalization. This form is concerned with discovering regularities and formulating new concepts and rules characterizing collections of any entities (objects, events, processes, etc.). It produces statements such as "most people are honest," "whenever there are independent events, the normal distribution should hold," or "John is in the habit of amblin' down to the soda fountain every day about now."

Examples of research on this topic are two programs by Lenat (15,23): AM, which searches for and develops new "interesting" concepts after being given a set of heuristic rules and initial concepts in elementary mathematics and set theory, and EURISKO, which formulates new heuristics. Another example is the BACON system (e.g., Ref. 28), which synthesizes mathematical expressions representing chemical or physical laws on the basis of given empirical data.

In the AI literature the term "concept learning" is frequently used in a more narrow sense than it is here, namely, to mean solely learning concepts from examples. One reason for this is historical, as this strategy was studied first, and most is known about it. It subsequently served as the springboard for studies of other strategies, but it continues to be the area most intensively investigated. Learning from examples and learning from observation and discovery (i.e., inductive learning in general) are fundamental forms of concept learning. When acquiring any abstract concept, examples are typically needed to achieve a deeper understanding of the concept; and initial learning of any concepts and natural laws is typically achieved by generalizing from our sensory observations. For these reasons the remainder of this entry concentrates on inductive learning. For coverage of other strategies the reader is advised to consult other references, in particular Ref. 29. The nature of inductive inference, which is the core of inductive learning processes, is explored in more detail.

Inductive Inferences Generates Hypotheses from Facts and/or Other Hypotheses

Inductive inference is the primary vehicle for creating new knowledge and predicting future events. It is usually characterized as reasoning from specific to general, from particular to universal, or from part to whole. Such a characterization is simple but not too informative. It does not identify all the components playing a role in the inductive process, nor does it explain how this inference is possible. To understand this inference more precisely, its major components are distinguished, and the properties of its conclusions are specified.

Given:

premise statements (facts, specific observations, intermediate generalizations) that provide information about some objects, phenomena, processes, and so on;

a *tentative inductive assertion*, which is an a priori hypothesis held about the objects in the premise statements (in some acts of inductive inference there may not be any tentative hypothesis; if there is such a hypothesis, the inductive process may be simplified, as it may involve merely a modification of the tentative hypothesis rather than creating a new hypothesis from scratch); and

background knowledge, which contains general and domain-specific concepts for interpreting the premises and inference rules relevant to the task of inference; it includes previously learned concepts, domain constraints, causality relations, assumptions about the premise statements and candidate hypotheses, goals for inference, and methods for evaluating the candidate hypotheses from these goals' viewpoints (specifically, the preference criterion or bias).

Determine:

an *inductive assertion* (a *hypothesis*) that strongly or weakly implies the premise statements in the context of background knowledge and is most preferable among all other such hypotheses.

A hypothesis strongly implies premise statements in the context of background knowledge if by using background knowledge (and standard rules of inference), the premise statements can be shown to be a logical consequence of the hypothesis. In other words, the assertion

Hypothesis & Background knowledge $\Rightarrow$ *Premise statements*

is valid, that is, true under all interpretations (the symbol $\Rightarrow$ denotes implication). A hypothesis that satisfies this condition is called a strong candidate hypothesis. In contrast, a weak hypothesis is the one that only weakly implies premise statements, that is, these statements are a plausible, but not certain, consequence of the hypothesis. The following two-part example illustrates both types of hypotheses.

Example: Part 1.

Premise statements:

Socrates was Greek. Aristotle was Greek. Plato was Greek.

Background knowledge:

Socrates, Aristotle, and Plato were philosophers. They lived in antiquity.

Philosophers are people. Greeks are people.

Preference Criterion. Prefer the hypothesis that is short and useful for deciding the nationality of philosophers.

Candidate hypotheses (a selection):

1. Philosophers who lived in antiquity were Greek.
2. All philosophers are Greek.
3. All people are Greek.

Preferred hypothesis:

4. All philosophers are Greek. (It is shorter than 1 and more specific than 3; it allows one, unlike 1, to determine the nationality of all philosophers.)

It can be seen that the original premise statements are a logical consequence of the generated hypothesis and background knowledge. The fact that the generated hypothesis is too general is a result of the poverty of the background knowledge and/or the premise assertions.

Example: Part 2. Suppose that the stock of facts has been enlarged with statements such as "Spencer was British" and "Hume was British" and that the background knowledge includes also the statement "Hume and Spencer were philosophers."

In this case a strong candidate hypothesis would be "All philosophers were Greek except Spencer or Hume, who were British." A weak hypothesis would be "Most (or some) philosophers were Greek." Given a fact that Plato was a philosopher, the new hypothesis, in contrast to the old one, does not allow one to conclude strongly that he was Greek. It allows one only to say that it is likely (or that it is possible) that he was Greek. However, unlike the first hypothesis, it will also not conclude strongly that philosopher Russell was Greek!

This example illustrates important properties of inductive inference. One is that it may not be truth preserving, that is, its conclusions may be incorrect though the premise statements are correct. Going back to the first hypothesis, though Socrates, Aristotle, and Plato were Greek, it certainly does not follow that all philosophers were Greek. This quality of nontruth preservation contrasts inductive inference with truth-preserving deductive inference. Figure 1 illustrates the relationship between deductive and inductive inference.

Inductive inference that produces strong hypotheses is falsity preserving. This means that if the original premise statements are false, the derived hypothesis will be false also. For example, if it were not true that Socrates was Greek, then clearly the first hypothesis, "All philosophers were Greek," could not be true either. Hypotheses generated by inductive inference have unknown truth status. They must be tested and verified before they become rules or accepted theories (see section on hypothesis verification).

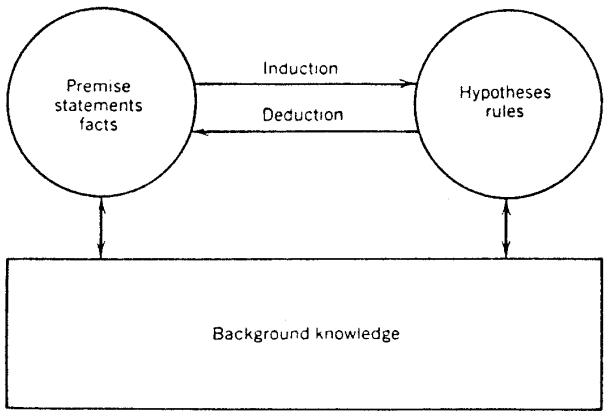

Figure 1. Relation between deduction and induction.

The premise statements, background knowledge, and derived hypotheses need to be expressed in some language. In human inference it is the language of the mind, a "mentalese," that at the surface level takes the form of natural language augmented with special representations of sensory stimuli, such as drawings, pictures, sounds, or gestures. In machine inference it is a formal language, such as propositional logic, predicate calculus or other logic-style formalisms, or a knowledge representation system, such as semantic networks, mathematical expressions, frames, scripts, or conceptual structures (30). Sometimes expressing the premise statements is easier in one language and expressing hypotheses is easier in another language.

In concept learning from examples (concept acquisition) the main concern is with a special case of inductive inference, called inductive generalization. Here both the premise statements and the hypothesis are either interpretable as descriptions of sets (in this case there is instance-to-class generalization) or as descriptions of components of some object or process (in the latter case there is part-to-whole generalization).

In instance-to-class generalization properties known to hold for a set of objects are assigned to a larger set of objects. This form can be seen in the example above, in which a property (the nationality) assigned by premise statements to a few individuals was assigned to all individuals in some class (all philosophers). In part-to-whole generalization the premise statements describe parts of some object, and the goal is to hypothesize a description of the whole object. For example, the following is a part-to-whole generalization.

Premise: His hands and his legs are strong.
Background knowledge: Hands and legs are parts of a body.
Hypothesis: His whole body is strong.

An important form of part-to-whole generalization is sequence or process prediction (31,32).

Inductive inference was defined as a process of generating descriptions that imply original facts in the context of background knowledge. Such a general definition includes inductive generalization and abduction as special cases. The term "abduction" was coined by the American logician Peirce (33). In abduction, the generated descriptions are specific assertions implying the facts (in the context of background knowledge) rather than generalizations of them. For example, given a premise assertion, "these roses are purple," and background knowledge "all roses in Adam's garden are purple," an abductive assertion would be "perhaps these roses are from Adam's garden."

A description that implies some facts can be viewed as an explanation of these facts. The most interesting form of an explanation is when it provides a causal, goal-oriented characterization of the facts. To derive such an explanation, background knowledge must contain, along with other inference rules, causal inference rules as well as a specification of the

goal(s) of inference. Generating causal explanations can thus be viewed as a form of inductive inference.

Inductive Inference Can Be Performed by Rules

One of the important results of research on inductive inference is the development of the concept of an inductive inference rule. An inductive inference rule performs some elementary act of inductive inference. It takes one or more assertions and generates an assertion that tautologically implies them. The concept of an inductive inference rule permits one to view inductive inference, at least conceptually, as a rule-guided process that starts with initial premises and background knowledge and ends with an inductive assertion (34). Here are a few examples of such rules:

Dropping conditions (removing a conjunctively linked condition from a statement; e.g., replacing the statement "a nation is strong if it has a strong economy and high determination" by "a nation is strong if it has high determination").

Turning constants into variables (e.g., it generalizes the statement "this apple tastes good" into "all apples taste good").

Adding options (it generalizes a statement by adding a disjunctively linked condition; e.g., it might generalize the statement "peace will be preserved if all nations have peaceful intentions" into "peace will be preserved if all nations have peaceful intentions or if nonaggressive nations are much stronger than the aggressive ones").

Climbing generalization tree (replacing a less general term by a more general term in a statement; e.g., generalizing the statement "I like oranges" into "I like citrus fruits").

A systematic presentation of inductive rules is in Ref. 34.

Instance Space versus Description Space

Earlier two forms of inductive learning have been distinguished: learning from examples and learning by observation. Learning a concept from examples is a process of constructing a representation of a designated class of entities by observing only selected members of that class and optionally nonmembers (counterexamples). Learning from observations involves creating concepts as useful classes for characterizing observations or any given facts. Both processes depend on the learner's background knowledge, in particular, on the type of description language the learner uses for characterizing examples and learned concepts.

In this context it is instructive to distinguish between an instance space and a description space. The instance space consists of all possible examples and counterexamples of concepts to be learned. Actually observed positive and negative examples constitute subsets of such an instance space. The description space is the set of all descriptions of instances or classes of instances that are possible using the description language specified by the learner's background knowledge. Learning a concept involves an interaction between the two spaces. Such an interaction may involve reformulation or transformation of initial assertions as well as experimentation and active selection of training examples (Fig. 2).

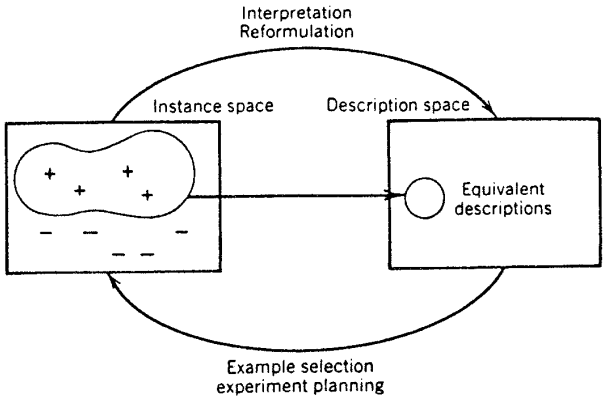

Figure 2. Interaction between instance space and description space.

Consider a simple case where examples of a concept (positive examples) and counterexamples (negative examples) are represented by attribute vectors, that is, by lists of values of certain attributes. Considering attributes as dimensions spanning a multidimensional space, each example maps to a point in this space. Points that do not correspond to any observed example represent potential examples. Such a space is called a feature space or an event space and can be viewed as a geometric model of an instance space.

One may ask where the attributes come from. In simple methods the attributes are defined by the teacher. Such methods are called selective because the learned concept does not include any new attributes but only those defined by a teacher. In more sophisticated methods the system is provided with some initial attributes plus various rules of inference, heuristics, or procedures that a learner uses for generating new attributes. The latter methods are called constructive (34,35).

Different subsets of the instance space correspond to different concepts. Descriptions of those concepts are elements of the description space. For simplicity, assume that the description space is the set of all logical expressions involving attributes used in characterizing examples. Depending on the constraints imposed on these expressions, all (or only some) subsets of the instance space can be represented by an expression in this language. Usually, any concept corresponds to a subset of (logically equivalent) descriptions in the description space.

A concept is consistent with regard to the examples if it covers some or all positive examples and none of the negative examples. A concept description is complete with regard to the examples if it covers all positive examples. A description of a concept that is both complete and consistent with regard to all examples is a candidate hypothesis. The requirement for completeness and consistency follows from the assumption that the hypothesis should imply the initial examples (see Ref. 34). The set of all candidate hypotheses is called the candidate hypothesis space or the version space. The candidate hypothesis space can be partially ordered by the relation of generality that reflects the set inclusion relation between the corresponding concepts. The most general hypothesis describes the concept that is the complement of the union of negative examples; and the most specific hypothesis describes the concept that is the union of all positive examples.

Because the candidate hypothesis space is usually quite

large, a preference criterion is used to decide which candidate hypothesis to choose. Such a criterion may favor, for example, hypotheses that are short, hypotheses that require the least effort to measure the attributes involved, or generally, hypotheses that best reflect the goal of learning.

If the concept representation language is incomplete, for example, allows one to express only conjunctive hypotheses, and a sufficient number of positive and negative examples is supplied, the resulting version space may contain only one candidate hypothesis. In such a case the preference criterion is not needed (17).

In summary, learning a concept can be described as a heuristic (qv) search (qv) through the description space for a most preferred hypothesis among all those that are consistent and complete with regard to the training examples.

Selected Methods of Inductive Learning

An important characteristic of learning methods is the way in which descriptions in the description space are generated and/or searched in relation to the examples or facts in the instance space. Three types of methods can be distinguished: data driven, model driven, and mixed. A data-driven method starts with selecting one or more examples, formulates a hypothesis explaining them, and then generalizes (and occasionally specializes) the hypothesis to explain further examples. A model-driven method starts with some very general hypotheses and then specializes (and occasionally generalizes) them to fit all the examples. Roughly speaking, data-driven methods proceed from specific to general, and model-driven methods proceed from general to specific. A mixed method has elements of both: It uses an example(s) to jump to one or more general hypotheses, tests the hypotheses, and then modifies them to fit other examples. Data-driven methods tend to be more efficient, and model-driven methods tend to be more tolerant of errors in data (29). Below are examples of the three types of methods.

Data-Driven Methods

Winston's Block World: Learning by Incremental Generalization and Modification. Winston's program (36) is an excellent representative of a data-driven method of concept learning. It learns structural descriptions of concepts in a blocks world (e.g., the concept of an arch) from representative examples and counterexamples provided by a teacher. The program represents examples and concepts in the form of a semantic network. At each step of learning it maintains only one working hypothesis. In searching for the final hypothesis, it uses a simple form of best-first search method. The basic algorithm can be described as follows:

1. Take first positive example of the concept and assume that it is a concept description.
2. If the next example is positive and does not satisfy the current concept description, generalize the description so that it includes the example.
3. If the next example is negative and satisfies the current description, specialize the description so that it excludes the example.
4. Repeat steps 2 and 3 until the process converges on a stable concept description.

The generalization step (step 2) applies such operators as dropping conditions, turning constants to variables, or climbing generalization tree. When confronted with multiple choice in generalizing, the program chooses the least "drastic" change to the current concept description. For example, it will replace a less general term by a more general term rather than drop a term. The specialization step (step 3) adds more conditions and introduces exceptions or the must-not conditions to the currently held hypothesis. There are usually many ways to specialize a hypothesis so that it does not cover a given negative example (as many as there are differences between the example and the hypothesis). For that reason the program favors the near misses, that is, negative examples that differ from the hypothesis in only a few or, in the best case, in only one aspect.

Other examples of data-driven methods are the candidate elimination algorithm (17,37) for learning from examples and the method for learning from observation embodied in the BACON system (28). The latter method discovers equations characterizing empirical laws.

Model-Driven Methods

Learning by Incremental Specialization and Modification: The *Meta-DENDRAL* Program. This program implements a model-driven method for discovering rules characterizing the operation of a mass spectrometer (38). These so-called cleavage rules predict which bonds in a molecular structure of a chemical compound will likely break when bombarded by electrons in the mass spectrometer. To avoid undue technical details of the specific domain, the rule-learning process is presented at a level of abstraction.

This process consists of two phases. First, the rule generation phase conducts a general-to-specific search of the space of possible cleavage rules (subprogram RULEGEN). Next, the rule modification phase makes the rules so obtained more precise and less redundant by performing local hill-climbing searches (subprogram RULEMOD). Training examples can be viewed as attribute vector descriptions of the environment of individual bonds in a molecule. Among the attributes are the type of atoms on both sides of the bond, the number of hydrogen and nonhydrogen atoms bound to each atom, number of unsaturated valence electrons of the atom, and so on. With each example is associated a decision as to whether the corresponding bond will break in the mass spectrometer. An important feature of this application is a large-sized, error-laden set of input examples.

The rule generation phase starts with the most general rule, stating that every bond will break. Abstracting from the specific domain-dependent notation, such a rule can be written:

If a bond is any bond, then it will break.

The next step specializes the left side of the parent rule by making a change to atoms at a specified distance from the bond. A change may involve changing properties of an atom or adding a new atom. New rules so obtained are then tested to see if they perform better in predicting the breaks in the given set of examples. This two-step process of rule specialization and testing repeats until a local optimum of performance is achieved. The resulting rules can be characterized as:

If a bond environment has properties so and so, then it will break.

Meta-DENDRAL was an important learning system that worked well in a real-world domain with noisy data. In addition to the process of rule development, outlined above, it also performed a sophisticated transformation of the initial data (the input spectrum) to usable training instances (the bond environment descriptions). In all aspects of its operation the program relied on a large amount of domain-specific knowledge.

Another example of a model-driven method is the concept-learning program, CSL (3), and its modified version, ID3 (39). The program starts by attempting to find the best one-attribute rule characterizing given examples. If this is not possible, it builds a decision tree of such rules that classifies all input examples. In such a tree nodes correspond to attributes, emanating branches to the attribute values, and leaves to classes.

Mixed Methods

Learning by Rapid Generalization and Stepwise Specialization: AQ11. Inductive concept learning can be viewed as a generate-and-test process. The "generate" part creates or modifies hypotheses and the "test" part tests how well the hypotheses fit the data. In data-driven methods the "generate" part is sophisticated and the "test" part is simple, whereas in model-driven methods the opposite holds. A mixed method, implemented in the program AQ11, attempts to more equally emphasize the "generate" and "test" parts.

AQ11 is a multipurpose learning program that formulates general rules describing various classes of examples (40). Input to the program consists of attribute value vector descriptions of examples from different classes. It also includes background knowledge about the application domain and a hypothesis preference criterion. The output can be viewed as rules,

$$\text{Condition} \Rightarrow \text{class}$$

where "condition" may be conjunction, or a disjunction of conjunctions, such that it describes all entities assigned to "class." A simplified version of the algorithm, called AQ, which underlies the nonincremental learning part of the program is as follows.

1. Select at random one positive example (called the seed).
2. Comparing the seed with the first negative example, generate all maximally general hypotheses that cover the seed and exclude the negative example.
3. Specialize the hypotheses to exclude all negative examples. This is done by considering one negative example at a time and adding, whenever neccessary, additional constraints to the hypotheses. After each step of specialization the newly generated hypotheses are ranked according to how well they classify remaining examples and according to other aspects defined in the preference criterion. Only the most promising hypotheses are kept. The set of hypotheses obtained at the end of the specialization process is called a star.
4. Select from the star the best-ranked hypothesis. If this hypothesis covers all positive examples, exit (a solution has been found). Otherwise, find positive examples that remain uncovered.
5. Repeat steps 1–4 for the remainder set. Continue until all positive examples are covered. The disjunction of hypotheses selected at the end of each cycle is a consistent and complete description of all the positive examples and maximizes the preference criterion.

Thus, the program builds a disjunctive description of a concept when a conjunctive description is not possible. The individual conjuncts in such a disjunction may significantly differ as to the size of coverage of the training examples. This allows for an interesting interpretation: The conjunct that covers most of the events could be viewed as a characterization of the typical, or "ideal," members and those with light coverage as a characterization of exceptional cases.

The incremental part of the program performs operations of modifying generated descriptions to fit new examples. The background knowledge of the program contains information about the properties of the attributes used to describe examples and various domain constraints. The program has been applied to various problems in medicine, agriculture, chess, and other areas. A more advanced version of the program, INDUCE (34), is capable of learning not only attribute-based but also structure-based concept descriptions. These descriptions characterize concepts as structures of components bound by various relationships, and are expressed in an extended predicate calculus. The program has the ability to utilize general and domain-specific knowledge to generate new attributes.

How are Learned Concepts Validated?

Although inductive inference represents the basic method for acquiring knowledge about the world and is one of the most common forms of inference, it suffers from a fundamental weakness. Except for special cases, results of this inference are inherently insusceptible to complete validation. This is because an inductively acquired hypothesis may have an infinite number of consequences, but only a finite number of tests can be performed. This property of inductive inference was observed early on by the Scottish philosopher David Hume and subsequently analyzed by twentieth-century thinkers such as Popper (e.g., Ref. 41). Consequently, one typically assumes that concept descriptions learned inductively have only a tentative status. When new examples become available, these descriptions are tested on them and, if neccessary, appropriately modified. A standard method for testing inductively acquired descriptions (rules) is to apply them to testing examples and compute a confusion matrix. Such a matrix records the number of correct and incorrect classifications of the testing examples by the rules.

Extended Notions of a Concept

The basic ideas and a few selected methods of concept learning have been described here. These methods were based on the notion that concepts are classes of entities describable by a logic-style description. This means that concept descriptions have sharp boundaries and all members are equal representatives of a concept. As pointed out above, this simplification,

though useful for research, misses some important aspects of the human notion of a concept.

Human concepts, except for special cases occurring predominantly in science (concepts such as a triangle, a prime number, a vertebrate, etc.), are structures with flexible and/or imprecise boundaries. They allow a varying degree of match between them and observed instances and have context-dependent meaning. Flexible boundaries make it possible to "fit" the meaning of a concept to changing situations and to avoid precision when not needed or not possible. The varying degree of match reflects the varying representativeness of a concept by different instances. Instances of a concept are rarely homogeneous. Among instances of a concept, people usually distinguish a "typical instance," a "nontypical instance," or, generally, they rank instances according to their typicality. By the use of context, the meaning of almost any concept can be expanded in a multitude of directions that cannot be predicted in advance. An imaginative discussion of this property is by Hofstadter (42), who shows how a seemingly well-defined concept, such as "First Lady," can express a great variety of meanings depending on the context in which it is applied.

Despite various efforts, the issue of how to represent concepts in such a rich and context-dependent sense remains open. This issue is, of course, crucial for concept learning because to learn concepts, the learner must be able to represent them. In view of this, a brief review of basic approaches to concept representation may be useful for understanding the current research limitations and directions in concept learning.

Smith and Medin (43) distinguish between three approaches: the classical view, the probabilistic view, and the examplar view. The classical view assumes that concepts are representable by features that are singly necessary and jointly sufficient to define a concept. This view is a special case of the one assumed in this entry, as it does not allow disjunctive concept descriptions.

The probabilistic view represents concepts as weighted, additive combinations of features. Using the aforementioned notion of a feature space, this means that concepts should correspond to linearly separable subareas in such a space. Experiments indicate, however, that this may be too limiting a view (43). The exemplary view represents concepts by one or more typical exemplars rather than by generalized descriptions.

The notion of typicality can be captured by a measure, called family resemblance. This measure represents the sum of frequencies with which different features occur in different subsets of a superordinate concept, such as furniture, vehicle, and so on. The individual subsets are represented by typical members. Nontypical members are viewed as corruptions of the typical, differing from them in various small aspects, as children differ from their parents (e.g., Refs. 44 and 45).

Another approach uses the notion of a fuzzy set as a formal model of a concept (46). Members of such a set are characterized by a gradual numerical set membership function rather than by the in–out function seen in the classical notion of a set. This set membership function is defined by people describing the concept and thus is subjective. This approach allows one to express the varying degree of membership of entities in a concept but does not have mechanisms for expressing the context dependence of the concept meaning.

Elements of the above approaches have been unified in a more recent idea, which postulates that the concept is characterized by a well-defined description, but the use of this description is flexible (47). If an entity does not satisfy the description precisely, a consonance degree is computed that specifies the degree to which the description is satisfied. Thus, objects precisely satisfying the formal description can be considered as typical concept members and those that satisfy approximately as less typical, with the degree of membership defined by the consonance degree. In the case of disjunctive descriptions the component (conjunction) that explains most of the examples can be viewed as representing the ideal form of a concept. Other components then represent exceptional cases. The method of computing consonance degree can be shared by many concepts; therefore, there is no need for storing a set membership function with each concept, as in the case of fuzzy sets. The dependencies among the attributes characterizing a concept and its relationship to other concepts can be expressed in the same logic-based formalism. Thus, in such a "flexible logic" approach the total meaning of a concept is distributed between its formal description and the function evaluating the degree of consonance. The description gives the basic meaning to a concept, and the evaluation function allows for its flexibility. Major questions, then, are how to properly distribute the concept meaning between these two components and how to express context-dependent meaning.

An adequate concept representation should include not only a description that permits one to recognize the given concept among other concepts or to evaluate the typicality of its members but also a number of other components. It should specify the constraints and correlations among the defining or characteristic attributes, the relationship of the concept to other concepts, its typical and nontypical examples, the dependence of meaning on different contexts, the purpose and use of the concept, and its position and role in knowledge structures and theories in which it is embedded. Many of these components are present in the representation described in Ref. 48. Murphy and Medin (24) argue that the role a concept plays in a theory that uses it provides a basis for conceptual coherence, that is, for explaining why certain classes of entities constitute a meaningful concept and some others do not. Further progress on concept learning is predicated on progress in concept representation.

Conclusion

Concept learning has been presented as a process of constructing a concept representation on the basis of information provided by an external source, a teacher, or an environment. The type of transformation performed by the learner defines the learning strategy. The main emphasis of this entry is on inductive learning, which is divided into learning from examples and learning from observation and discovery. Principles are described that underly inductive inference, and several methods are presented for concept learning from examples.

A number of topics in concept learning have not been covered. Among these are methods for creating new concepts, noninductive learning strategies, techniques for evaluating learned concept descriptions, and learning from noisy or incompletely defined examples. The general references include papers on these topics.

BIBLIOGRAPHY

1. C. I. Hoveland, A "communication analysis" of concept learning, *Psychol. Rev.* 59(6), 461–472, 1952.
2. J. S. Bruner, J. J. Goodnow, and G. A. Austin, *A Study of Thinking*, Wiley, New York, 1956.
3. E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in Induction*, Academic Press, New York, 1966.
4. A. Newell, J. C. Shaw, and H. A. Simon, A Variety of Intelligent Learning in the General Problem Solver, Rand Corporation Technical Report, Santa Monica, CA, 1959.
5. A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Dev.* (3), 210–229, 1959, reprinted in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*, McGraw-Hill, New York, 1963, pp. 71–105.
6. M. Kochen, "Experimental study of 'Hypothesis Formation' by Computer," in C. Cherry (ed.), *Information Theory: 4th London Symposium*, Butterworth, London and Washington, DC, 1961.
7. S. Amarel, On the Automatic Formation of a Computer Program which Represents a Theory, in M. Yovits, G. Jacobi, and G. Goldstein (eds.), *Self-Organizing Systems*, Spartan Books, Washington, DC, 1962, pp. 107–175.
8. R. B. Banerji, Computer Programs for the Generation of New Concepts from Old Ones, Neure Ergebnisse der Kybernetik, in K. Steinbuch and S. Wagner (eds.), Oldenberg-Verlag, Munich, 1964, p. 336.
9. N. Bongard, *Pattern Recognition*, Spartan Books, New York, 1970 (translation from a Russian original published in 1966).
10. S. Watanabe, *Pattern Recognition as an Inductive Process, Methodologies of Pattern Recognition*, Academic Press, New York, 1968.
11. M. Minsky and S. Papert, *Perceptrons*, MIT Press, Cambridge, MA, 1969.
12. P. H. Winston, Learning Structural Descriptions from Examples, Ph.D. Thesis, Report No. TR-231, AI Laboratory, MIT, 1970 (reprinted in *The Psychology of Computer Vision*, P. H. Winston (ed.), McGraw-Hill, New York, 1975.
13. B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg, A Heuristic Programming Study of Theory Formation in Sciences, *Proc. of the Second International Joint Conference on Artificial Intelligence*, London, 1971, pp. 40–48.
14. R. S. Michalski, A Variable-Valued Logic System as Applied to Picture Description and Recognition, in F. Nake, A. Rosenfeld (eds.), *Graphic Languages*, North-Holland, Amsterdam, 1972, pp. 20–47.
15. D. B. Lenat, AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search, Ph.D. Dissertation, Stanford University, 1976.
16. P. Langley, "BACON: A Production System that Discovers Empirical Laws," *Proc. of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977, pp. 344–346.
17. T. M. Mitchell, Version Spaces: A Candidate Elimination Approach to Rule Learning, *Proc. of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, August 1977, pp. 305–310.
18. P. H. Winston, "Learning and reasoning by analogy," *CACM* 23:12), 689–703 (1979).
19. J. R. Anderson, A Theory of Language Acquisition Based on General Learning Principles, *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia, August 1981, pp. 97–103.
20. R. S. Michalski and R. E. Stepp, "Learning from observation: Conceptual clustering," in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 331–364.
21. R. C. Schank, Looking at Learning, *Proceedings of the European Conference on Artificial Intelligence*, Orsay, France, July 1982, pp. 11–18.
22. J. G. Carbonell, Learning by Analogy: Formulating and Generalizing Plans from Past Experience, in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, 1983, pp. 137–161.
23. D. B. Lenat, The Role of Heuristics in Learning by Discovery: Three Case Studies, in R. S. Michalski, J. G. Carbonell, T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, 1983, pp. 243–306.
24. G. L. Murphy and D. L. Medin, "The role of theories in conceptual coherence," *Psychol. Rev.* 92(3), 289–316 (1985).
25. N. Hass and G. G. Hendrix, Learning by Being Told: Acquiring Knowledge for Information Management, in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 305–427.
26. D. Michie, "Memo functions and machine learning," *Nature* 218(5136), 19–22 (1968).
27. T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli, "Explanation-Based Generalization: A Unifying View," *Machine Learning* 1(1), 47–80 (1986).
28. P. Langley and G. L. Bradshaw, Rediscovering Chemistry with the Bacon System, in R. S. Michalski, J. G. Carbonell, T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 307–329.
29. T. G. Dietterich and R. S. Michalski, A Comparative Review of Selected Methods for Learning from Examples, in R. S. Michalski, J. G. Carbonell and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 41–81.
30. J. F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA 1984.
31. H. A. Simon and G. Lea, Problem Solving and Rule Induction: A Unified View, L. W. Gregg, (ed.), in *Knowledge and Cognition*, Lawrence Erlbaum, Potomac, MD, pp. 105–127, 1974.
32. T. Dietterich and R. S. Michalski, Learning to Predict Sequences, in R. S. Michalski, J. G. Carbonell and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, Morgan Kaufman, Los Altos, CA, 1986, pp. 63–106.
33. C. S. Peirce, *Essays in the Philosophy of Science*, The Liberal Arts Press, New York, 1957.
34. R. S. Michalski, Theory and Methodology of Inductive Learning, in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 83–134.
35. L. A. Rendell, Substantial Constructive Induction: Feature Formation in Search, *Proc. of the Ninth IJCAI*, Los Angeles, CA, August 1985, pp. 650–658.
36. P. H. Winston, "Learning Structural Descriptions from Examples," *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975, ch. 5.
37. T. M. Mitchell, P. E. Utgoff, and R. Banerji, Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics, in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 163–190.
38. B. G. Buchanan and E. A. Feigenbaum, "Dendral and Meta-Dendral: Their applications dimension," *Artif. Intell.* 11, 5–24 (1978).
39. J. R. Quinlan, Learning Efficient Classification Procedures and their Application to Chess End Games, in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning: An Artifi-*

cial Intelligence Approach, Tioga, Palo Alto, CA, 1983, pp. 463–482.

40. R. S. Michalski and J. B. Larson, Selection of Most Representative Training Examples and Incremental Generation of VL1 Hypotheses: The Underlying Methodology and a Description of Programs ESEL and AQ11, Report 867, Department of Computer Science, University of Illinois, Urbana, 1978.
41. K. R. Popper, *Objective Knowledge: An Evolutionary Approach*, Oxford, Clarendon Press, 1979.
42. D. R. Hofstadter, *Metamagical Themas: Questing for the Essence of Mind and Pattern*, Basic Books, New York, 1985, Chapter 24.
43. E. E. Smith and D. L. Medin, *Categories and Concepts*, Harvard University Press, Cambridge, MA, 1981.
44. L. Wittgenstein, *Tractatus Logico-Philosophicus*, Routledge & Kegan Paul, London, 1921.
45. E. Rosch and C. B. Mervis, "Family resemblances: Studies in the internal structure of categories," *Cog. Psychol.*, 7(4), 573–605 (1975).
46. L. A. Zadeh, "A Fuzzy-algorithmic approach to the definition of complex or imprecise concepts," *Int. J. Man-Machine Stud.* 8(3), 249–291 (1976).
47. R. S. Michalski and R. L. Chilausky, "Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis," *Pol. Anal. Inform. Sys.* 4(2), 125–161 (June 1980).
48. D. Lenat, M. Prakash, and M. Shepherd, "CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks," *AI Mag.* 6(4), 65–85 (1986).

General References

T. G. Dietterich, B. London, K. Clarkson, and G. Dromey, Learning and Inductive Inference, in P. R. Cohen and E. A. Feigenbaum (eds.), *Handbook of Artificial Intelligence*, Vol. 3, 325–511, W. Kaufmann, Los Altos, CA, 1982.

J. McCarthy, Programs with Common Sense, *Proceedings of the Symposium on the Mechanization of Thought Processes*, Vol. 1, National Physical Laboratory, 1958.

N. Zagoruiko, *Empirical Prediction Algorithms, Computer Oriented Learning Processes*, J. C. Simon (ed.), Noordhoff, Leiden, The Netherlands, 1976.

R. S. MICHALSKI
University of Illinois

This work was supported in part by the NSF under grant No. DCR 84-06801, by the ONR under grant No. N00014-82-K-0186, and by DARPA under grant No. N00014-K-85-0878.

CONCEPTUAL DEPENDENCY

Conceptual dependency (CD) is a theory of natural language and of natural-language processing (see Natural-language generation; Natural-language understanding). It has been developed by Schank with the motivation to enhance one's ability to construct computer programs that can understand language well enough to summarize it, translate it into another language, and answer questions about it. At the heart of the theory lies the conjecture that language is a medium whose purpose is communication. Therefore, the central issue dealt with by the theory is the kinds of things that can be communicated, the meaning content of the communication.

What inferences are made?
When are these inferences made?
Where do they come from?

For example, most people would agree that the sentence "John sold his old car" contains a reference to money even though the word "money" is not mentioned in the sentence. Furthermore, most people would agree that as a consequence of John's action, he no longer owns that car. Any computer program that understands this sentence must answer no to the question "Does John own the car?" and yes to the question "Did John receive money?"

How could a program know that? To model language understanding on a computer, one needs a strong theory of human inference that operates on the level of conceptual manipulations. Furthermore, in order for a theory of language to have relevance in the field of AI, it must provide a representation of meaning as well as the means to map into and out of that representation (see Representation, knowledge).

Conceptual dependency theory is a theory of the representation of meaning. It is a representation of everyday concepts and events in a way that reflects natural thinking and communication about those concepts and events. At the time of its development, the approach taken by Schank was not considered unusual within the AI framework. Since AI is largely an experimental field, the theory and its computer implementations were viewed as investigation into the dynamics of natural-language understanding. However, in the field of linguistics thoughts about the nature and the purpose of language were oriented in a direction opposite to that reflected by Schank's theory, and the latter was considered radical.

Conceptual Structures

Conceptual dependency theory views understanding of natural language as a process of mapping linear strings of words into well-formed conceptual structures. A conceptual structure is defined as a network of concepts, where certain classes of concepts can be related in specific ways to other classes of concepts (see also Semantic networks). The basic axiom of the theory is:

> For any two sentences that are identical in meaning, regardless of language, there should be only one representation.

A corollary that derives from it is:

> Any information in the sentence that is implicit must be made explicit in the representation of the meaning of that sentence.

The rules by which classes of objects combine may be viewed as conceptual syntax rules. It is important to note that these rules underly the language, but they are independent of it. They are rules of thought as opposed to rules of a language. The initial framework consists of the following rules (1):

> The meaning of a linguistic proposition is called a conceptualization or CD form.
>
> A conceptualization can be active or stative.
>
> An active conceptualization consists of the following slots: actor; action; object; and direction, source (from) destination (to) (instrument).