KNOWLEDGE BASED SYSTEMS AND NEURAL NETS

R. Moreno-Díaz and **R. Moreno-Díaz Jr.**
Institute of Cybernetics, University of Las Palmas de Gran Canaria, Spain

Keywords*:* Artificial Intelligence, Knowledge Engineering, Expert Systems, Formal Neural Nets, Neural Models, Artificial Neural Nets

Contents

Summary

Knowledge–based systems are here introduced according recent conceptions by separating knowledge representation methods from the methods for knowledge use. In each case, three fundamental levels (the user level, the knowledge engineering level and the tool development level) are distinguished. Neural Nets are classified according their function as models for logical machines, as tools for modelling living systems and as a paradigm for parallel distributed computation in learning systems. Illustrations and applications are presented in all cases.

1. Introduction

The basic concepts of knowledge-based systems (KBS) imply the separation of knowledge representation itself from the way it is used. There are three levels in KBS: the level of the user, the level of the knowledge engineer and the level of the tool developments. There are a variety of ways to represent knowledge, which range from logic, to frames, production systems and other. KBSs have been dominated by applications in medicine, though practically all areas of applied knowledge, which combine heuristics with formal methods of knowledge acquisition and reasoning, have been areas of interest.

Neural Nets are classified into three types:

- Formal Neural Nets, which are biological counterparts of logical systems (propositional logic, deterministic, probabilistic and fuzzy automata), of interest to find connectivistic counterparts for automata.
- Natural Neural Nets modeling, which uses tools ranging from differential equations to symbolic processing.
- Artificial Neural Nets are a class of computing paradigm, the so-called connectivistic approach, in which computation is carried out by a set of similar units. They aim is to make computation faster, cheaper and more reliable. They have the capacity of learning.

2. Knowledge-based Systems

2.1. Definitions and Scope

A knowledge-based system (KBS) is a set of software packages capable of supporting explicit representations of knowledge on a specific domain of expertise and capable of using it to provide solutions to problems in that domain. The solution aimed at must be the same that would be reached by an expert in the domain.

There are various fundamental concepts which distinguish KBSs from conventional software programs to solve algorithmic problems (e.g. problems in transmission networks, or in engineering computation structures). First, there is the separation of knowledge from how to use it; second, the use of highly specific domain knowledge and third, the heuristic rather than algorithmic nature of the knowledge employed.

The first idea goes back to the General Problem Solver (GPS) developed in the late 1950s and early 1960s. GPS solved a problem by finding a sequence of operators that eliminated the difference between the initial state of a problem and one of the desired goal states. These operators represented general knowledge about what operations were possible within the domain. GPS used general search techniques to find the right sequence of operators that would progressively decrease the distance between the original state and the goal state of a problem.

The second underlying concept was initially exploited in the development of the expert system DENDRAL and Meta-DENDRAL (late 1960s and early 1970s). DENDRAL operation is for inferring the molecular structure of unknown compounds from mass spectral data while Meta-DENDRAL assists the scientist in determining the dependence of mass spectrometric fragmentation on substructural features, that is, a highly specific domain of knowledge.

The concept of using heuristic (i.e. experience-based) knowledge arose from the aim to duplicate the human capacity to solve problems without continuously having to resort to models or algorithms. This seemed to be the link between the early researches in AI and the new field of knowledge-based systems. The development of MYCIN at Stanford in the 1970s used these concepts. MYCIN is an expert (KBS) developed to help diagnosing and specifying treatments in blood disorders through a "conversation" with a physician. The system directs the conversation by asking questions about the signs and symptoms of the patient and by requesting the performance of certain laboratory tests.

After the diagnosis and the prognosis of the patient are established, the treatment protocol is advised.

In the 1980s it became apparent that focusing on the development of typical artificial intelligence systems (development of formalisms, inference mechanisms and tools) to implement KBSs—though it seemed rather promising—failed at large. This was because developments could not provide for the transfer from the academy to commercial use to build large KBSs. This has been the recent goal of the so-called knowledge engineering (KE) approach to the implementation of KBSs.

In the early 1980s the development of a KBS was seen as a transfer process of human knowledge (knowledge acquisition) to form a knowledge base. This transfer was based on the assumption that the required knowledge already existed, and, typically, it was acquired by interviewing experts on how they solved specific tasks. Because of many inconveniences and limitations of this transfer approach, a kind of shift was happening towards the consideration of building KBSs by the so- called modeling approach, in which the acquisition process is no longer seen as a transfer of knowledge to an appropriate computer representation, but as a process of constructing a model of the domain.

2.2. KBS Levels

In the field of KBSs, three levels can be easily recognized: the level of the end-user or client, the level of the knowledge engineer and the level of tools (also called “shells”) builders.

For the end-users to which a KBS system is addressed, the system consists essentially of three components: an intelligent domain-specific software, the user interface and a problem-specific data base.

The intelligent software is for the user simply a black box which operates with some logic, but which derives appropriate results. The user interface must provide for convenient interaction using menus, specialized parts of natural language, graphical displays and pictures that can be used by the software to state questions and to provide for explanations of its conclusions under the query of the user.

A good design of the user interface to facilitate communication is typically a must in practical KBS, since, as a rule, the end-user is not a computer expert. Third, there is the problem-specific database, which contains all the information provided by the user to initially define the problem: the data and information provided while the intelligent software is proceeding its operation, as well as the conclusions on each step of the path to ultimate solutions or final conclusions. The database must contain all the data, which, from the user viewpoint, describes the problem at hand; data which, in some cases, can be provided by automated sensor readings or by any other data management system.

The knowledge engineer must define the type of knowledge, its organization, its representation and the way to handle it to understand the data that will be provided by the user and proceed towards a solution. This is achieved by close interaction with an expert in the domain.

On the level of a KE specialist, the intelligent software can be classically viewed as composed of two subsystems, a domain-dependent knowledge base and the so-called inference engine. The success of current expert system technology is, in large part, due to a decision to enforce a strict separation between the two, such that the inference engine, capable of reasoning with the knowledge in the base, is strictly domain independent.

The inference engine is the knowledge interpreter stored in the knowledge base, so that from its own contents and the data of the problem to be solved, it derives additional data and conclusions.

On the level of tool development, the problem is to build a software package that contains all the features needed by the KE to build the system. These software packages are known as shells, of which there are a number commercially available, from very general ones to special purpose.

2.3. Knowledge Representation and Knowledge Acquisition

In KBSs, knowledge representation consists in expressing knowledge (facts, rules) in a formalism that can be treated by a computer program. Several methods of knowledge representation have been developed. Usually, a knowledge base contains several types of them. Typical ways of representing knowledge include the following:

Logical representation uses predicate logic, Horn-clauses or functional representations. Associative networks are rather popular ways of representation. Their name denotes the association of different objects of the knowledge by labeled arcs. This permits very efficient access to the different properties of an object. It is in fact an object-oriented representation.

Frames reflect the situation that facts and properties are many times related, even though they are not easily recognized to belong together. Thus, a frame is constructed, which is given a name which, when called, evokes the content of subparts, called slots, that are to be filled with values. Frames are also a form of object oriented representation. They are also support for automated knowledge acquisition, since the system may call for the slots to be filled once the frame is evoked. Frames also contribute to the efficiency of knowledge processing, since they permit to concentrate within small regions of the base.

One particularly successful way of knowledge representation is the procedural, rule-based or production system. The system is formed by a working memory, a set of production rules and a rule interpreter. The working memory consists of a set of data that represents the present state of the system. Production rules are of the conditional form if (condition) then (action). The conditions are stated in the form of logical formulae of some literal representing events. The rule interpreter is encharged in verifying the conditions, that is, matches the conditions to the entries in the working memory. If the matching is successful, the specified action is executed, which leads to a new state of the system.

To control the behavior of the rule interpreter, the control mechanism may provoke a forward chaining, where the process starts with the entries in present state of the working memory and proceed to execute rules until a desired goal is achieved. In backward chaining, the process starts with the goal and searches rules back until the conditions match with the state in working memory.

There are several other important ways to model knowledge for its representation. Extensions of first order logic permit the so-called modal logic, where modalities are expressed by operators to express necessity and possibility. Another relevant subject is the representation of evidence, certainty or uncertainty, by either adding extra parameters to predicates, or introducing new logical operators, such as the "fuzzy" quantifiers.

For visual knowledge representation there has been developed a range of tools from more or less geometric-like representations at lower level ends of visual processing to highly semantic approaches at the level of image understanding.

Even though the development of inference engines has shown continuos progress, so that KBS designers have a variety of available shell systems, it is still the case that knowledge acquisition is the bottleneck in KE systems development. The development and maintenance of knowledge bases continue to be major problems in KBSs.

That is why, as it was pointed before, there is a shifting from the traditional paradigm expert-knowledge engineer-knowledge base (E-KE-KB paradigm), where the knowledge engineer is promoting the transfer of knowledge from the expert to the knowledge base, towards systems in which the role of the engineer is less and less relevant in the creation and maintenance of the knowledge base. Anyway, there are still applications where the E-KE-KB paradigm, provides for good results, though large KBSs require for some automated acquisition system.

At the other end of the line, there are research efforts based in similarity-learning which tend to eliminate the presence of engineers and experts, by for example, learning the rules from a library of previously solved problems. This tendency is inverse to what happened in visual recognition learning systems, where the efficiency gained by injecting non-learned knowledge into the system has been notorious.

In KBSs, the omission of the engineer in the acquisition paradigm, leads to the model-based paradigm, in which knowledge acquisition is guided by the level of domain knowledge currently existing in a computer acquisition system. The computer acquisition system houses a model of the domain. Models of the domains run from partial models (like the syntactic model Teiresias, developed in 1982–84), semantic models and causal models, more recently being explored. The road ahead in knowledge acquisition seems to be oriented towards the expert-computer acquisition system-knowledge base (E-CAS-KB) paradigm.

-
-
-

Bibliography

Adeli H., ed. (1990). *Knowledge Engineering*, Vol. I. Fundamentals. New York: McGraw-Hill. [Contains 10 chapters by different authors which provide for a very clear and rigorous exposition of different subjects from knowledge acquisition and representation to planning, machine learning, logic and expert systems.]

González A. J. and Dankel D. D. (1993). *The Engineering of Knowledge-Based Systems*. Englewood Cliffs, NJ: Prentice Hall. [This is a very clear introductory text on knowledge-based systems.]

Guida G. and Tasso G. (1994). *Design and Development of Knowledge Based System*. Chichester: John Wiley and Sons. [A practical handbook for KBSs project leaders, designer and knowledge engineers.]

Mira-Mira J. and Sánchez-Andrés V., eds. (1999). *Engineering Applications of Bio-Inspired Artificial Networks*. Lecture Notes in Computer Science No. 1607. Berlin: Springer. [It is the second volume of papers from the bi-annual IWANN Conference. It contains latest applications of ANNs and other connectivistic approaches. Volume 1, on natural nets, entitled *Foundations and Tools for Neural Modeling* (no. 1606) shows great part of what are the trends in Natural Neural Nets Modeling.]

Moreno-Díaz R. (1995). Natural and srtificial nets. In *Genetic Algorithms in Engineering and Computer Science* (Winter, Périaux, Galán, Cuesta, eds.), pp. 83–110. Chichester: John Wiley and Sons. [A summarized presentation of the evolution and main contributions in the subject.]

Moreno-Díaz R. and Mira-Mira J., eds. (1995). *Brain Processes, Theories and Models*. Cambridge, MA: MIT Press. [A collection of papers on Natural Neural Nets modeling and Artificial Neural Nets, from a conference in honor of W.S. McCulloch; it covers the fields quite well.]

Russell S. and Norvig, P. (1995). *Artificial Intelligence, A Modern Approach*. Englewood Cliffs, NI: Prentice-Hall. [An excellent text, which covers practically all subjects included in Artificial Intelligence. The Chapter on Neural Nets provides for a rapid, clear and good view.]

Studer R., Benjamins W. R., and Fensel D. (1998). Knowledge engineering: principles and methods. *Data and Knowledge Engineering* **25**, 161–197. [A very good account of the modern bases of the field as well as visions for future developments.]

Biographical Sketches

Roberto Moreno Diaz holds a Degree in Physics (1962) and a PhD (1965) on Logical Neural Nets and Electronic Models of Neurons, both from the University of Madrid (now Complutense). From 1965 to 1968 he was a research staff at the Charles Stark Draper Laboratory of the Technological Institute of Massachussetts, and later consultant for the same, under the supervision of W.S. McCulloch. Since 1968 he is full Professor, first of Electromagnetics at the University of Zaragoza and later of Computer Sciences at the University of Las Palmas de G.C..Is author or coauthor of over one hundred papers on neurocybernetics, retinal theory and natural and artificial vision. Has organised eleven International Conventions on Computer Sciences, Computer Aided Systems Theory and Neurocybernetics. Is coeditor of 12 volumes on these subjects, published by various houses including Alianza Editorial, Springer-Verlag, Hemisphere and the MIT Press. He is Academician Correspondent for the Royal Academy of Exact, Physical and Natural Sciences of Madrid since 1981 and Founder Member and Vice-president of the Canary Academy of Sciences. He has been invited lecturer in various Universities of Europe and

USA. Presently, he is Director of the University Institute for Cybernetics Research and Technologies, Universidad de Las Palmas de G.C.

Roberto Moreno Díaz Jr, Assistant Professor at the Universidad de Las Palmas de G.C., obtained his Degree in Physics from the Universidad Complutense de Madrid (1989), Spain, and his PhD in Computer Science from the Universidad de Las Palmas (1993). He has developed several research stays at the State University of New York at Buffalo, USA, and the Universities of Vienna and Johannes Kepler-Linz, Austria. Specialized in bioinspired artificial vision and formal models of natural visual processing, he is author or coauthor of approximately 30 papers that have been published in several international journals and is main author of a book on computational models of visual perception (2000, in spanish). As research associate at the Canary Islands Institute of Technology (1995-97) has participated in several R+D projects funded by the Canary Islands Government and the Spanish Commission for Science and Technology. Having supervised two PhD Thesis on Computer Science (1997 and 2001), he teaches Mathematical Methods for Computer Science and Biocibernetics in the Faculty of Informatics and currently is the secretary of the University Institute for Cybernetics Research and Technologies.