EXPERT SYSTEMS

UNIT 9

Introduction

The development of expert systems could be viewed as the single most important accomplishment of artificial intelligence.

They emerged in the 1970s at a time when the whole field of artificial intelligence was subdued and AI was criticized for not producing real-time, real-world working systems.

The goals of early systems, were to develop general and powerful methods that could be applied to diverse problem-solving areas.

An expert system is a computer program that simulates the thought process of a human expert to solve complex decision problems in a specific domain.

An expert system uses both facts and heuristics to solve difficult decision problems based on knowledge acquired from an expert

Introduction

An expert system may be viewed as a computer simulation of a human expert.

Early programs, such as DENDRAL proved weak in terms of their power of generality.

The behavior of the best general problem solvers we know, human problem-solvers, is weak and shallow, except in those areas where the human problem-solver is a specialist.

Contrary to early beliefs, most humans are experts in only their own specialized fields; they do not possess some kind of magic that enables them to quickly generate the finest and most cogent set of rules for an arbitrary problem domain

What we do know is that a long training is required before any particular domain can be mastered.

Introduction

There are five stages of skill acquisition in the progression from novice to expert

1. Novice
   - The novice just follows rules and has no coherent understanding of the task domain.
   - There is no context to the rules, no understanding, just the ability to follow rules to accomplish a task. for example – following some instructions to assemble a new product, or typing in a computer program from a paper copy.

2. Advanced Beginner
   - The **advanced beginner** starts to learn more from experience and is able to employ contextual clues.
   - For example, when learning to make coffee with a coffee machine, we follow the instructions, however, we also use our sense of smell to tell us when the coffee is indeed ready.
   - In other words, we learn from clues that we can perceive in the task environment.

Introduction

3. Competent

- The **competent** skill performer no longer needs to just follow rules, but also has a clear understanding of the task environment. He is able to make decisions, draw upon a hierarchy of rules, and to recognize patterns.
- Competent performers might be goal-oriented and can alter their behavior according to conditions.
- For example, a competent driver knows how to alter his driving according to weather conditions, including speed, the gear, windshield wipers, mirrors, and so forth.
- At this point, the performer will have developed intuition or know-how.
- Performer at this level is still analytical, able to combine elements to make the best decisions based on his experience.

Introduction

4. Proficient
   - The **proficient** problem-solver will not only be able to recognize what the situation and appropriate choices are, but will also be able to deliberate the best way to implement a solution.
   - An example is the doctor who knows what a patient's signs and symptoms suggest, and will carefully consider what the possible choices of treatment are.

5. Expert
   - The **expert** "generally knows what to do, based on mature and practiced understanding."
   - When the expert copes with his environment, he does not see problems as detached from his effort to solve them, nor does he worry about the future and devise elaborate plans.
   - "*When things are proceeding normally, experts don't solve problems and don't make decisions; they do what normally works*" i.e. "those who are proficient or expert make judgments based upon their prior concrete experiences in a manner that defies explanation"
   - "*experts act irrationally*"; that is, action without conscious analytic decomposition and recombination.

Expert Systems: Characteristics

1. Domain Specific

Expert systems are typically very domain specific.

For example, a diagnostic expert system for troubleshooting computers must actually perform all the necessary data manipulation as a human expert would .

2. Special programming languages

Expert systems are typically written in special programming languages.

The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process.

3. Operates as an interactive system

Expert systems responds to questions, asks for clarifications, makes recommendations and aids the decision making process

Expert Systems: Characteristics

4. Tools have ability to sift (filter) knowledge

Storage and retrieval of knowledge

Mechanisms to expand and update knowledge base on a continuing basis.

5. Make logical inferences based on knowledge stored

Simple reasoning mechanisms is used

Knowledge base must have means of exploiting the knowledge stored, else it is useless;

e.g., Learning all the words in a language, without how to combine those words to form a meaningful sentence.

6. Ability to explain reasoning

Remembers logical chain of reasoning; therefore user may ask

- for explanation of recommendation
- factors considered in recommendation

Enhances user confidence in recommendation and acceptance of expert systems.

Expert Systems: Characteristics

Capabilities to assign confidence values

- Can deliver quantitative information
- Can interpret qualitatively derived values
- Can address imprecise and incomplete data through assignment of confidence values.

Cost effective alternative to human expert

- Expert systems have become increasingly popular because of their specialization, albeit in a narrow field.
- Encoding and storing the domain specific knowledge is economic process due to small size.
- Specialists in many areas are rare and the cost of consulting them is high; an expert system of those areas can be useful and cost effective

Knowledge Engineering

Knowledge, which often arrive in a crude, inexact, incomplete, and poorly specified form is the key to the power of any expert system.

Like human amateurs, experts do not develop instantly, but must be built incrementally over time.

Knowledge will be inexact for probabilistic sciences such as medicine, geology, weather, and certain other disciplines, yet the techniques for propagating uncertainties have been highly developed and expert systems can do this much more systematically, quickly, and accurately than humans can.

Perhaps surprisingly, human experts often find it difficult to express the logic, intuition, and heuristics they use to analyze data when it comes under their management.

For example the professor of mechanics and the unicyclist both are fine doing what they do expertly.

However, once they try to understand and explain their expertise, their performance dramatically degenerates. The unicyclist can’t explain her abilities and, likewise, the professor’s knowledge of the laws of mechanics won’t make him a successful unicyclist!

Expert Systems: Components

Inference engine

An inference engine implements the reasoning process of artificial intelligence; which is an identification of human reasoning.

The purpose of an inference engine is to extract information and from the database for providing the answers, predictions, and suggestions just like a human expert.

Knowledge acquisition

The knowledge acquisition is responsible for providing the knowledge to the database in an expert system

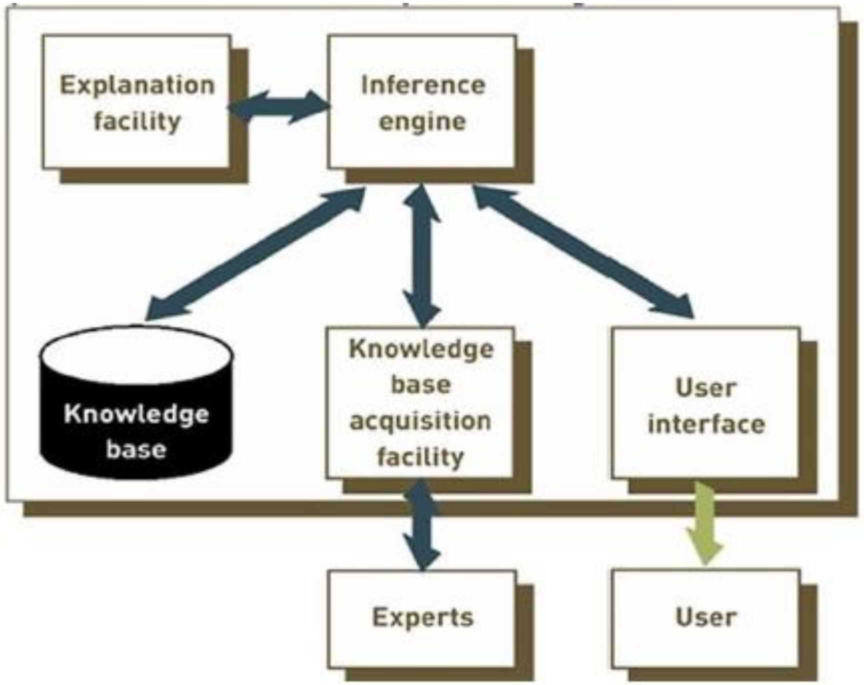

Fig. Components of expert system

Expert Systems: Components

Explanation facility

The explanation facility provides a particular solution by showing a path to the user in order to reach a certain conclusion

The user can know how the expert system arrived at the solution, why some alternatives were rejected, why some information was asked for etc.

User interface

The user interface manages the dialogue between the user and the system. It provides facilities such as menus, graphical interface, etc

Knowledge base.

consists of problem solving rules procedures, and intrinsic data relevant to the problem domain.

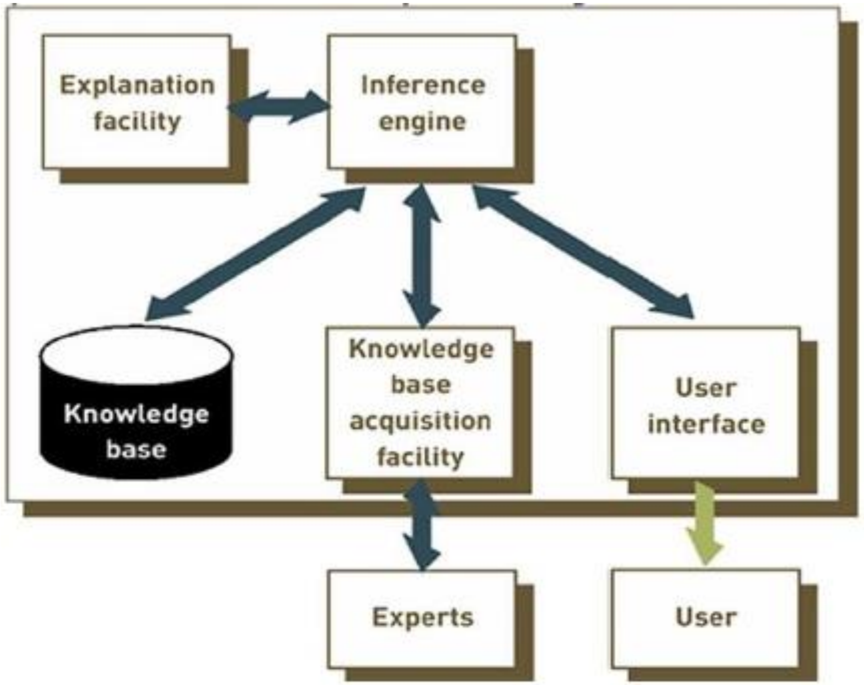

Fig. Components of expert system

Expert Systems

AI systems, particularly production systems and expert systems based on them, are distinct from traditional computer science programs because they tend to separate the computational components from the knowledge based components.

Hence, in terms of expert systems, the inference engine is distinct from the knowledge base.

The database of expert systems typically comprises rules “invoked by pattern-match with features of the task-environment which can be added to, modified, or deleted by the user.”

Databases of this type are called knowledge bases. Users can employ knowledge bases in three typically distinct fashions:

1. Getting answers to problems – *user as client*

2. Improving or increasing the system’s knowledge – *user as tutor*

3. Harvesting the knowledge base for human use – *user as pupil*

Expert Systems

The people who use expert systems to improving or increasing the system’s knowledge are known as **domain specialists**.

It is not possible to build an expert system without the help of a domain specialist.

The person who extracts the knowledge from a domain specialist and formulates it as a knowledge base is known as the knowledge engineer.

“The process of extracting the knowledge from the domain specialist’s head (a very important process) is known as **knowledge acquisition**.”

The process of constructing the knowledge base via a series of interactions between the domain specialist and the knowledge engineer is known as **knowledge engineering**.

Often this process will involve a number of iterations and refinements of the rules over time by the knowledge engineer as he becomes more familiar with the domain specialist’s rules.

Knowledge Acquisition

The task of producing the knowledge from the expert and organizing it into a usable system has always been viewed as a difficult one.

It is also most important to the power of an expert system and, in essence, represents the expert's understanding of the problem.

Formally, this process is called Knowledge acquisition, and it is the biggest challenge to building an expert system.

Knowledge elicitation can be a long and arduous task involving a number of tedious sessions.

These sessions could be in the form of interactive discussion with exchange of ideas or in the form of interviews or case studies.

In the latter form, the expert is observed as she tries to solve a real problem. Whatever the method, the goal is to uncover the expert's knowledge and gain better insight into her problem-solving skills.

People have wondered why the expert cannot simply be probed by questions for her knowledge

Knowledge Acquisition

Following are the characteristics of experts which is the reason for difficulty in knowledge acquisition:

1. They tend to be very specialized in their domain and will tend to use language that is domain specific.

2. They have largely heuristic knowledge, which is uncertain and imprecise.

3. They have difficulties in expressing themselves.

4. They bring to bear many sources of knowledge to achieve their performance.

The process of converting *shallow knowledge* (which might be based on intuition) into *deep* or *deeper knowledge* (which might be hidden in the expert's subconscious) is called the k**nowledge compilation problem**.

Knowledge-engineering paradox: "The more competent domain experts become, the less able they are to describe the knowledge they used to solve problems!"

CLASSIC EXPERT SYSTEMS: DENDRAL

Heuristic DENDRAL (later shortened to DENDRAL) was a chemical-analysis expert system. The substance to be analyzed might, for example, be a complicated compound of carbon, hydrogen, and nitrogen.

Starting from spectrographic data obtained from the substance, DENDRAL would hypothesize the substance's molecular structure.

DENDRAL would quickly reduce from hundreds of possible structures to several or possibly one.

If several possible structures were generated, then they would be listed with probabilities attached.

DENDRAL demonstrated that computers could perform on a par with human experts in a restrictive domain.

In chemistry it performed on a par or above a PhD chemist.

The program was largely written in a dialect of Lisp called Interlisp.

Subroutines such as CONGEN were written in Fortran and Sail.

It was widely marketed and used by chemists throughout the United States.

CLASSIC EXPERT SYSTEMS: MYCIN

MYCIN is the most written about, studied, and modeled program.

This rule-based expert system was designed for diagnosis and therapy recommendations for infectious blood diseases caused by bacteria in the blood and meningitis (bacterial disease that causes inflammation of the membrane surrounding the brain and spinal cord). Such diseases can be quickly fatal if not treated early.

MYCIN required some 20-person years to develop, uses backward chaining, and is comprised of more than 400 rules.

To prove its conclusion and be able to explain it, MYCIN works backward through its rule base, searching for confirming evidence. Using backward chaining, MYCIN could perform on a par with human experts.

MYCIN was eventually used as a training program for medical interns.

As a model, it is exemplary of many good features and reasons for why one would like to build an expert system.

MYCIN employs probabilities, it has an explanation facility, it tries to communicate in a friendly and useful manner for physicians.

CLASSIC EXPERT SYSTEMS: EMYCIN

MYCIN proved to be such a successful expert system that it was determined that it should be generalized.

William van Melle used the MYCIN inference engine and a 1975 Pontiac Service Manual to build a 15-rule system for diagnosing problems with the car horn circuit.

This toy system provided the basis for the development of the first expert system shell, EMYCIN. EMYCIN stands for “Essential” or “Empty” MYCIN.

A shell is a special-purpose tool designed for certain types of applications in which the user must supply only the knowledge base.

In the case of EMYCIN the shell was made by removing the medical knowledge base of the MYCIN expert system.

The goal was naturally to retain the excellent features of MYCIN, including the representation of domain-specific knowledge, ability to traverse the knowledge base, the ability to support uncertainty, hypothetical reasoning, explanation facilities, and so forth.

EMYCIN supported both forward and backward chaining and led to the development of many expert systems, including PUFF an application for the diagnosis of pulmonary problems.

It was a very important development for expert-systems technology because it provided a tool whereby expert systems could be built “cost effectively”.

CASE BASED REASONING

"What is the purpose of history if we don't learn from it?"

Case-based reasoning (CBR), an approach to problem-solving that really is the basis for many of the fundamental ways civilized man functions and makes decisions.

The essence is that we learn from experience—the experiences others have had and our own experiences. It is on this basis that we make decisions.

Naturally, these experiences must somehow be documented, otherwise their usefulness is limited.

People make decisions based on their previous experiences

- For example, lawyers it is precedent. How have cases with similar circumstances in the past been resolved? What is the tendency of the particular judge we are dealing with? Is he conservative or liberal? What kind of decisions is the judge likely to make, based on the precedents in similar cases?
- Given the particular signs and symptoms that a patient has had, and given the patient's age, medical history, and other known relevant factors (e.g., existing conditions, previous surgeries, allergies to drugs, medical insurance) a doctor is able to make the decision(s) most likely to result in a favorable outcome.

CASE BASED REASONING

CBR is all about building AI systems that are able to match cases of solutions according to precedent; in other words, trying to solve new problems by matching them to solutions of old problems.

Hence, it is about building knowledge-based systems that learn from previous situations.

The main element of a cbr system is the case base; a structure that stores the problems, elements (cases), and its solutions.

Therefore, a **case base** can be visualized as a database where a collection of problems is stored, keeping a relationship with the solutions to every problem stored, which gives the system the ability to generalize in order to solve new problems.

The learning capabilities of cbr systems are defined by the result of their own structures, typically composed of four phases: **Retrieval, Reuse, Revision,** and **Retention**.

CASE BASED REASONING

Retrieve: finding the case most similar to the proposed problem, and retrieving it from the case base.

Reuse: Once a series of cases are extracted from the case base, the second phase adapts the selected case to it the current problem.

Revision: Once the system finds a solution to the problem it is revised and checked to see whether it is indeed a solution to the problem.

Retain: Once the proposed solution is confirmed as appropriate, it is retained and can serve as a solution to future problems.

CBR is not a new area of AI and is often used when the domain rules are incomplete, ill-defined, or inconsistent.

Case-based approaches can be useful in enabling an expert system to learn from previous experiences by storing solutions that have worked or failed in the past.

This can greatly shorten the overall problem-solving process.

END OF UNIT 9