IMPORTANT INSTRUCTIONS

1. The following Lectures already taught in face to face teaching and would be part of mid-term & final exam as well.

Week # 2 Lecture #3&4

ARTIFICIAL INTELLIGENCE INTRODUCTION

RELATED VIDEOS ON YOUTUBE			
Sr.	**Channel**	**Video Name**	**URL**
1	Dr Umair Abdullah	AI Dream (Final Goal)	https://youtu.be/nd59HQMId4A https://youtu.be/H2XdcfFr_-8
2	Dr Umair Abdullah	Introduction of Expert Systems	https://youtu.be/Ot6mOX5FOEUhttps://youtu.be/g4lJB_VBX4Ehttps://youtu.be/5FhGkd5CRy4 https://youtu.be/G5Sg4MLtiLw
3	Dr Umair Abdullah	Required Features of AI Languages	https://youtu.be/f3Pr1fwxAe4

Learning Objectives

1. Background: Human Nature
2. AI Dream
3. Introduction of Expert Systems
4. Characteristics of an Expert System
5. Comparison with Human Expert
6. Components of an Expert System
7. Expert system Examples
8. Expert System Shells
9. Required Features of AI Languages

Assignment #0

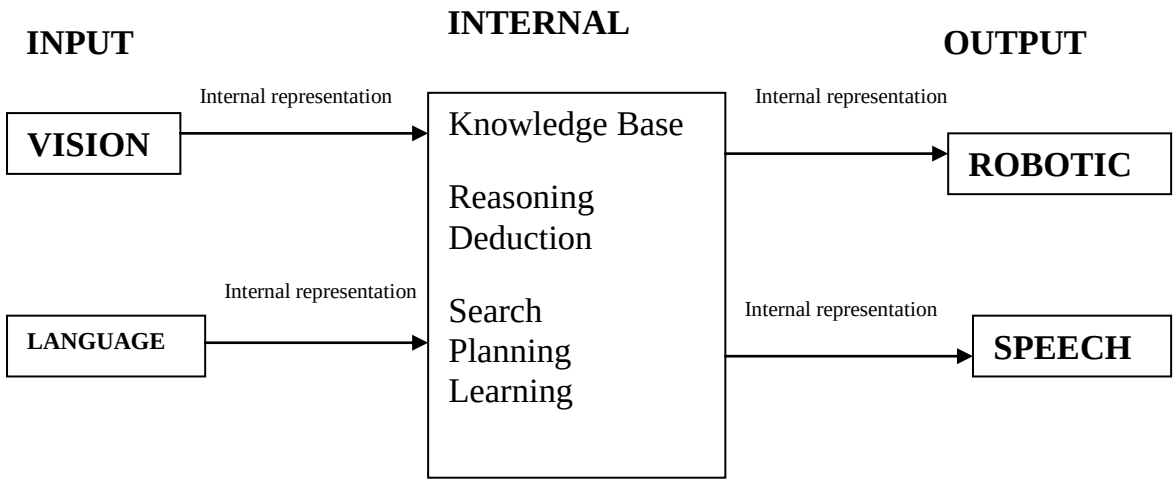
Dear students no assignment has been given with these lecture notes, as these topics had been discussed in face to face teaching at the institute. You may just go through the contents for founding the basis/ background of AI.

AI Dream (FINAL GOAL)

- Nature of Humans
- Invented many tool
- We have to operate the tools
- We want something to do our work
- Operating/ Monitoring of other tools
- Thinking/ reasoning/ decision-making
- Some labor work

We want to make artificial being so that he doeseverything for us and we have to do nothing (not even thinking or asking him to do). This is the dream of AI

Intelligent Being



Input and output are only for communication. Basic intelligence resides in the internals. It is the thinking of mind. Our course will also focus on these internals.

Point: Brain is hardware, mind is software.

What is knowledge, what do we do with knowledge. Stored experiences are knowledge. Break up the intelligence. For example1 burning of hand from Heater. Next time we take care of it.

We reason from already existing knowledge to solve the current problem. Bus hitting example 2. Reasoning is the way of getting inference from the premises. It can be right or wrong. Deduction is a type of reasoning in which we have complete explanation of everything and

conclusion drawn in this way is always correct. Example of burning of hand, what exactly happen when a hot heater touches the human hand, is it only for heater or for any other hot object?

Mind has to search the experiences which are relevant to the current problem. There are billions of experiences (of own and of others), mind use special searching techniques and immediately get the relevant experiences and then the inference engine of mind calculate the solution of the problem.

Example 3 how do we apply a formula, like solving a quadratic equation $x^2 + 3x + 40 = 2$. Mind will immediately recall the formula (if it is there). There is categorization of experiences.

Example 4 Pizza eating, going from BIIT to Pizza Hut. You will arrange money, transport, and company (of friends or family). Arranging coming events in proper sequence while considering their alternatives, is called planning. Learning is also involved in this process. It is necessary for the growth of intelligence and problem solving ability. Example 5 you are studying in class. It is acquiring new knowledge. Or you can say, digestion of experience (justification of new experience, linking new experience with the existing one). Several types of learning, Rote learning, explanation based learning, example base learning etc.

Vision and language are the two basic inputs of humans (and also of dreamed intelligent being). Although touch smell and taste are other three senses of humans but these are not as important as vision and language. Language is any meaningful pattern (sound or symbols). Vision is the most impressive input sense of human beings in which computers are still far behind. In computer science vision and language are the old areas of research. There are many systems which use visual techniques, like monitoring system, security systems (face and thumb recognition).

Robotics is also an old area. Honda, Sony, robots available which walks. Robotics is involved in manufacturing of electronic equipment and other things (automobiles). Three dimensional motion, robots are making robots.

Introduction of Expert Systems

Systems which behave like a human expert in a complex real life problem domains are known as expert systems. Human experts are able to perform at high level because they know a great deal about their areas of expertise. This simple observation is the underlying rationale for the design of knowledge-based problem solvers. An expert system, as these programs are often called, uses domain specific knowledge to provide “expert quality” performance in a problem domain. Generally, expert system designers acquire this knowledge with the help of human domain experts, and the system emulates their methodology and performance. As with skilled humans, expert systems tend to be specialist, focusing on a narrow set of problems. Also, like humans, their knowledge is both theoretical and practical: the human experts that provide the system’s knowledge have generally augmented their own theoretical understanding of the problem domain with tricks, shortcuts, and heuristics for using that knowledge they have gained through problem solving experience. Unlike a human, however, current programs cannot learn from their own experience: knowledge must be extracted from humans and encoded in a formal language. This is the major task facing designers of knowledge intensive problem solvers.

Expert systems should not be confused with cognitive modeling programs, which attempt to simulate human mental architecture in details. Expert system neither copy the structure of human mind, nor are they mechanism for general intelligence. They are practical programs that use heuristics strategies developed by humans to solve specific classes of problems.

Because of the heuristic, knowledge intensive nature of expert level problem solving expert systems generally:

1. Support inspection of their reasoning processes, both in presenting intermediate steps and in answering questions about the solution process.
2. Allow easy modification, both in adding and in deleting skills form the knowledge base.
3. Reason heuristically, using (often imperfect) knowledge to obtain useful problem solutions.

The reasoning of an expert system should be open to inspection, providing information about the state of its problem solving, and explanations of the choices and decisions that the program is making. Explanations are important for several reasons: first, if a human expert such as a doctor or an engineer is to accept a recommendation from the computer, he/she must

be satisfied the solution is correct. Indeed, human experts will accept advice from another human, let alone a machine, without understanding the justifications for it. This need to have answers explained is more than mistrust on the part of users: explanations help people relate the advice to their existing understanding of the domain and apply it in a more confidant and flexible manner.

Second, when a solution is open to inspection, we can evaluate every decision taken during the solution process, allowing for partial agreement and the addition of new information or rules to improve the performance. This help greatly in debugging the system and refining the knowledge base.

The exploratory nature of AI and expert system programming requires that programs be easily prototyped, tested, and changed. AI programming languages and environments are designed to support this iterative development methodology. In pure production system, for example, the modification of a single rule has no global syntactic side effects. Rules may be added or removed without requiring further changes to the larger programs. Expert system designers have commented that easy modification of the knowledge base is a major factor in producing a successful program.

The third feature of expert systems is their use of heuristic problem solving methods. Expert system designers have discovered, in formal “tricks of the trade” and “rules of thumb” are an essential complement to the standard theory presented in textbooks and classes. Sometimes these rules augment theoretical knowledge in understandable ways; often they are simply shortcuts that seem unrelated to the theory but have been empirically shown to work.

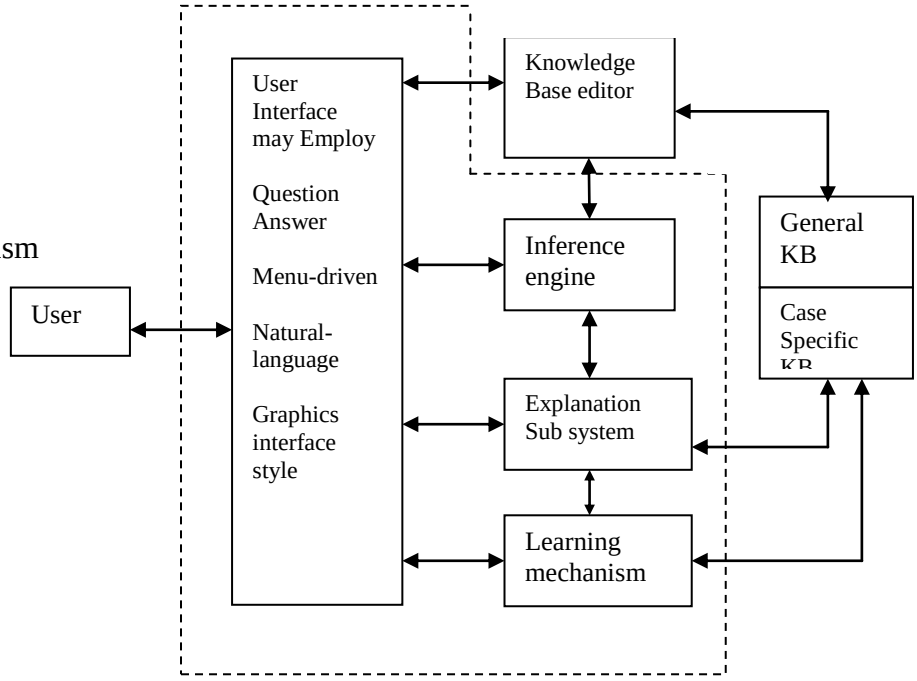
Expert systems have been built to solve a range of problems in domains such as medicine, mathematics, engineering, chemistry, geology, computer science, business, law, defense, and education. These programs have addressed a wide range of problem types; the following list is a useful summary of general expert system problem categories.

1. Interpretation — forming high level conclusions or descriptions from collections of raw data.
2. Prediction — projecting probable consequences of given situations.
3. Diagnosis — determining the cause of malfunctions in complex situations based on observable symptoms.

4. Design — finding a configuration of system components that meets performance goals while satisfying a set of design constraints.
5. Planning — devising a sequence of actions that will achieve a set of goals given certain starting conditions and runtime constraints.
6. Monitoring — comparing a system's observed behavior to its expected behavior.
7. Debugging and Repair — prescribing and implementing remedies for malfunctions.
8. Instruction — detecting and correcting deficiencies in students' understanding of a subject domain
9. Control — governing the behavior of a complex environment.

Components of an Expert System

- User Interface
- Knowledge Base
- Inference engine
- Explanation
- Learning mechanism



Comparison with Human Expert

- Permanent
- Easy to transfer
- Easy to document
- Consistent
- Affordable
- Easy to duplicate

Expert System Shells

- It is a software containing no knowledge (but have mechanism for representing the knowledge and Inferencing on it) used to develop expert systems. It is like an expert system containing all the modules except knowledge.
- Modules of expert system shells
- Knowledge representation mechanism
- Reasoning module
- Explanation module
- Knowledge editor (for adding new knowledge)
- Integrating mechanism
- Examples
  - OPS-5 (Official Production system) used to develop XCON
  - EMYCIN
  - EXSYS (for PC over 5000 rules can be created)
  - KEE (Knowledge Engineering Environment)

Characteristics of an Expert System

- Robustness
- Flexibility
- Accountability
- Soundness
- Reusability
- Extensibility

➢Robustness

System performance should not be dependent on idea circumstances (input). It should gracefully decrease in case of wrong input or missing information.

➢Flexibility

An expert system should be flexible in the sense that it should be able to perform a number of related tasks as well. And also in the sense that its decision should not be strict or limited.

➢Accountability

This is a unique characteristic of expert systems. That an expert system should be able to give clear justification of the decision made. In other words its manipulation of knowledge would be visible to the qualified user, so that wrong decision making could be prevented/removed.

➢Soundness

An expert system should have a thorough, well-defined, and justified knowledge of its problem domain. So that it could produce good expert level decisions.

➢Reusability and Extensibility

If possible, an expert system construction should be in such a way that it could be easily integrated with other systems. And also system should be extensible (for future enhancement) either by built in learning mechanism or by simple addition of code.

Expert system Examples

- DENDRAL
- MACSYMA
- MYCIN
- PROSPECTOR
- HEARSAY-II
- PUFF, GUIDON, NEOMYCIN
- HASP
- INTERNIST
- XCON
- DRILLING ADVISOR

Expert systems have not been around for very long. The following list places some of the best known systems in context:

- Beginnings (1965 – 1970) : DENDRAL, MACSYMA
- Prototypes (1970 - 1975) : MYCIN, PROSPECTOR, HEARSAY
- Experimenting (1975 – 1981): PUFF, GUIDON, INTERNIST
- Commercial Systems (1981- onwards): XCON, Drilling Advisor

DENDRAL's job was to help interpret mass spectrometer data. Imagine you have a mystery chemical, and electron beam can be used to smash the molecules, which tend to break at certain characteristic places. The bits are charged and electric and magnetic fields can be used to steer the fragments to various collection points, according to their weight and charge. The output of the spectrometer is essentially a histogram of how much of what weight was found. A complicating factor is that sundry of the bits tend to glue back together. DENDRAL applies various kinds of chemical knowledge to the combinatorial reasoning task of figuring out what the original chemical was.

MACSYMA is a set of tools for helping to manipulate collections of mathematical equations. It is still very much in use. By nuclear physicists who has large sets of simultaneous equations to solve.

MYCIN is the golden oldie of expert systems. Its job was to figure out, from simple and quickly obtainable data, what kind of bacterial infection a newly admitted hospital patient

might have and to recommend an appropriate set of antibiotics that will cover all the possibilities, keeping him alive and in a stable condition while the time consuming lab test were done. It only dealt with bacterial infection, and only of certain kinds. It is worth remembering that most of MYCIN's knowledge was based on heuristics derived from consensus of opinion at Stanford Medical School. So it is quite possible that MYCIN will perform poorly if tested against experts from some other medical establishments.

PROSPECTOR accepted data gathering by a field geologist, and advised about whether it was worth doing more expensive explorations at that site. It had one formal test in which it performed well. It had knowledge of only fifteen kinds of mineral deposits. Thus it was only and encouraging prototype. However a typical PROSPECTOR section cost only about ten dollars of computer time, and took just a few minutes. This would save a field geologist from waiting a few months for one of the few experts to look at his data. It is worth remembering that the average field geologist finds nothing useful in his working lifetime.

HEARSAY-II demonstrated that a machine might be able to understand fairly natural speech, on a highly constrained topic. It received input from a microphone and understood it fairly reliable and quickly.

MYCIN, PROSPECTOR, HEARSAY-II have radically different architectures; each has spawned a number of derivatives, first the techniques of each were applied in other areas to see if they still worked. Later, the specific knowledge was hollowed out and replaced by a "general purpose" knowledge representation mechanism, to produce what is called an *expert system shell.* Although there are many such shells are now available the representation language and facilities the hollowed out shells offer a not truly general each suits only a very limited range of applications, and can be pushed to handle a few more (such as EMYCIN, EXPERT).

PUFF advises about the causes of abstractive airway disease. It was originally implemented using EMYCIN a shell developed from MYCIN. It was eventually recreated in BASIC to run on an APPLE computer.

GUIDON was an attempt to use MYCIN's knowledge based for teaching MYCIN's diagnostic skills. It was not a great success. MYCIN knowledge was a behaviorist kind, with the stimuli being particular patterns in the data and the responses being the creation of new data. It didn't

have the kind of causal knowledge needed to be able to explain why certain conclusions might be expected to follow from certain data.

HASP process data from sonar arrays on the sea-bed, to maintain "current best view" about what kind of ships were in the vicinity and where they were heading. It was developed for the US Navy and worked well in the tests. It had an architecture based on that of HEARSAY-II.

INTERNIST was intended to handle all of internal medicine (a huge domain), and did cover nearly all of it. It mutated CADUCES, another monster system. Both have deficiencies but do work. They represent yet another flavor of knowledge representation.

NEOMYCIN was a later attempt to a MYCIN-like system, intended to cover a wider range of subject matter and to reason in a manner such closer to that actually employed by doctors – gathering some patient data. Forming some hypothesis, and then seeking to differentiate between those hypothesis by asking and collecting relevant data.

XCON was used digital equipment corporation to process customers' orders for VAX computers. Each order specifies a collection of ingredients from a catalogues, but it's easy to make mistakes. For example you may order six tape derives but forget to order the controller card or their cabinets, forget that you cannot get all six clustered around the CPU and so will need longer cables of some of them, forget that you may need a bigger power supply to handle the controller for them and so on. DEC people have been heard to say that this expert system saves them over 10 million dollars per year. And others have been heard to say that it hasn't saved any cost of staff, but that now the staff is kept happy maintaining and extending the expert system.

Drilling Advisor was produced by TeKnowledge a leading ES firm for Elf Aquitaine a large oil company. It offers advice to the drill team on an oil rig about what to do when the drill gets stuck. Bearing in mind that an oil rig may cost over 100,000 dollar per day to run and that it may take a whole day to remove a drill from the hole, anything that save some time is likely to be worthwhile.

Required features of AI languages

1. Scope of AI (Whole World, Mind-world) : Symbolic Manipulation
2. Dynamic & Flexible: Variables, Data Structures, Recursive
3. Pattern Matching (our faces,)
4. Data vs. Code : Learning Ability, Calculator Example
5. Nature of Humans (Functionality added at runtime)

.