Natural Language Processing

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be:

- Speech
- Written Text

Components of NLP

There are two components of NLP as given:

Natural Language Understanding (NLU)

Understanding involves the following tasks:

- Mapping the given input in natural language into useful representations.
- Analyzing different aspects of the language.

Natural Language Generation (NLG)

It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

It involves:

- **Text planning:** It includes retrieving the relevant content from knowledge base.
- **Sentence planning:** It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
- **Text Realization:** It is mapping sentence plan into sentence structure.

The NLU is harder than NLG.

Difficulties in NLU

- NL has an extremely rich form and structure.
- It is very ambiguous. There can be different levels of ambiguity:
    - **Lexical ambiguity:** It is at very primitive level such as word-level.

- For example, treating the word “board” as noun or verb?
- **Syntax Level ambiguity**: A sentence can be parsed in different ways.
- For example, “He lifted the beetle with red cap.” – Did he use cap to lift the beetle or he lifted a beetle that had red cap?
- **Referential ambiguity**: Referring to something using pronouns. For example, Rima went to Gauri. She said, “I am tired.” - Exactly who is tired?
- One input can mean different meanings.
- Many inputs can mean the same thing.

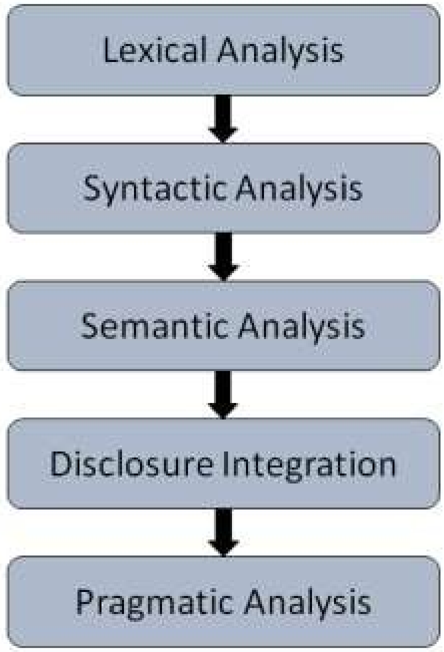
NLP Terminology

- **Phonology:** It is study of organizing sound systematically.
- **Morphology:** It is a study of construction of words from primitive meaningful units.
- **Morpheme:** It is primitive unit of meaning in a language.
- **Syntax:** It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.
- **Semantics:** It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.
- **Pragmatics:** It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.
- **Discourse**: It deals with how the immediately preceding sentence can affect the interpretation of the next sentence.
- **World Knowledge:** It includes the general knowledge about the world.

Steps in NLP

There are general five steps:

1. **Lexical Analysis**
   It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
2. **Syntactic Analysis (Parsing)**
   It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.



3. **Semantic Analysis**
   It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as “hot ice-cream”.
4. **Discourse Integration**
   The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
5. **Pragmatic Analysis**
   During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Implementation Aspects of Syntactic Analysis

There are a number of algorithms researchers have developed for syntactic analysis, but we consider only the following simple methods:

- Context-Free Grammar
- Top-Down Parser

Let us see them in detail:

Context-Free Grammar

It is the grammar that consists rules with a single symbol on the left-hand side of the rewrite rules. Let us create grammar to parse a sentence –

“The bird pecks the grains”

Articles (DET): a | an | the.

Nouns: bird | birds | grain | grains

Noun Phrase (NP): Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

Verbs: pecks | pecking | pecked

Verb Phrase (VP): NP V | V NP

Adjectives (ADJ): beautiful | small | chirping

The parse tree breaks down the sentence into structured parts so that the computer can easily understand and process it. In order for the parsing algorithm to construct this parse tree, a set of rewrite rules, which describe what tree structures are legal, need to be constructed.

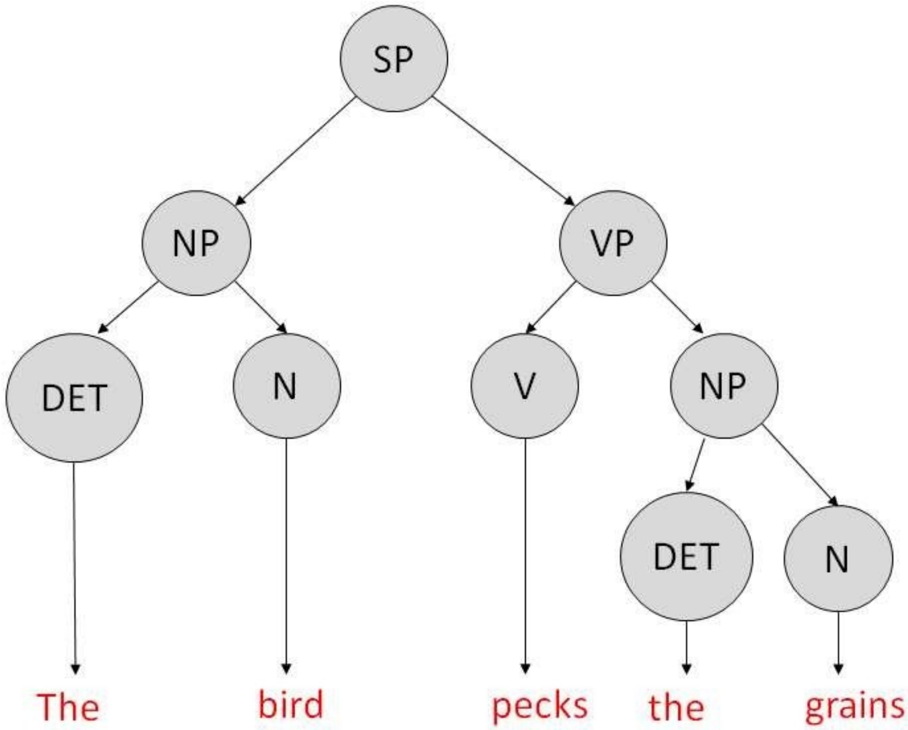
These rules say that a certain symbol may be expanded in the tree by a sequence of other symbols. According to first order logic rule, ff there are two strings Noun Phrase (NP) and Verb Phrase (VP), then the string combined by NP followed by VP is a sentence. The rewrite rules for the sentence are as follows:

S -> NP VP
NP -> DET N | DET ADJ N
VP -> V NP

Lexocon:
DET -> a | the
ADJ -> beautiful | perching
N -> bird | birds | grain | grains

V -> peck | pecks | pecking

The parse tree can be created as shown:



Now consider the above rewrite rules. Since V can be replaced by both, "peck" or "pecks", sentences such as "The bird peck the grains" can be wrongly permitted. i. e. the subject-verb agreement error is approved as correct.

Merit: The simplest style of grammar, therefore widely used one.

Demerits:

- They are not highly precise. For example, "The grains peck the bird", is a syntactically correct according to parser, but even if it makes no sense, parser takes it as a correct sentence.
- To bring out high precision, multiple sets of grammar need to be prepared. It may require a completely different sets of rules for parsing singular and plural variations, passive sentences, etc., which can lead to creation of huge set of rules that are unmanageable.

Top-Down Parser

Here, the parser starts with the S symbol and attempts to rewrite it into a sequence of terminal symbols that matches the classes of the words in the input sentence until it consists entirely of terminal symbols.

These are then checked with the input sentence to see if it matched. If not, the process is started over again with a different set of rules. This is repeated until a specific rule is found which describes the structure of the sentence.

Merit: It is simple to implement.

Demerits:

- It is inefficient, as the search process has to be repeated if an error occurs.
- Slow speed of working.

8. Expert Systems

Expert systems (ES) are one of the prominent research domains of AI. It is introduced by the researchers at Stanford University, Computer Science Department.

What are Expert Systems?

The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

Characteristics of Expert Systems

- High performance
- Understandable
- Reliable
- Highly responsive

Capabilities of Expert Systems

The expert systems are capable of:

- Advising
- Instructing and assisting human in decision making
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

They are incapable of:

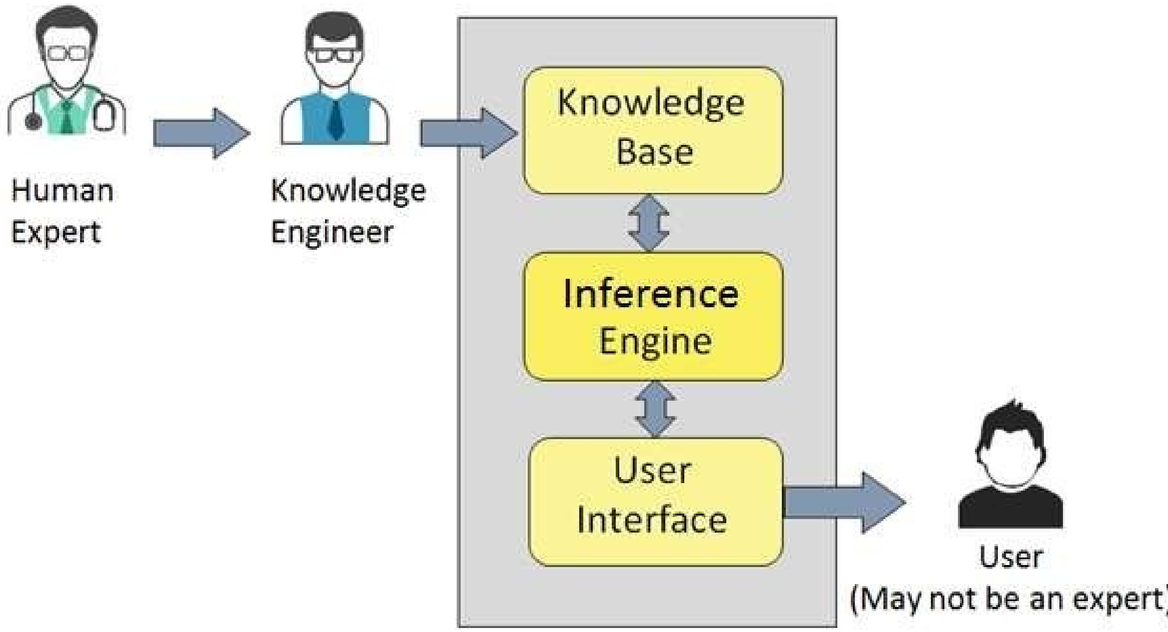
- Substituting human decision makers
- Possessing human capabilities
- Producing accurate output for inadequate knowledge base
- Refining their own knowledge

Components of Expert Systems

The components of ES include:

- Knowledge Base
- Inference Engine
- User Interface

Let us see them one by one briefly:



Knowledge Base

It contains domain-specific and high-quality knowledge.

Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge.

What is Knowledge?

The data is collection of facts. The information is organized as data and facts about the task domain. **Data**, **information**, and **past experience** combined together are termed as knowledge.

Components of Knowledge Base

The knowledge base of an ES is a store of both, factual and heuristic knowledge.

- **Factual Knowledge** – It is the information widely accepted by the Knowledge Engineers and scholars in the task domain.
- **Heuristic Knowledge** – It is about practice, accurate judgment, one's ability of evaluation, and guessing.

Knowledge representation

It is the method used to organize and formalize the knowledge in the knowledge base. It is in the form of IF-THEN-ELSE rules.

Knowledge Acquisition

The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base.

The knowledge base is formed by readings from various experts, scholars, and the **Knowledge Engineers**. The knowledge engineer is a person with the qualities of empathy, quick learning, and case analyzing skills.

He acquires information from subject expert by recording, interviewing, and observing him at work, etc. He then categorizes and organizes the information in a meaningful way, in the form of IF-THEN-ELSE rules, to be used by interference machine. The knowledge engineer also monitors the development of the ES.

Inference Engine

Use of efficient procedures and rules by the Inference Engine is essential in deducting a correct, flawless solution.

In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.

In case of rule based ES, it:

- Applies rules repeatedly to the facts, which are obtained from earlier rule application.
- Adds new knowledge into the knowledge base if required.
- Resolves rules conflict when multiple rules are applicable to a particular case

To recommend a solution, the inference engine uses the following strategies:

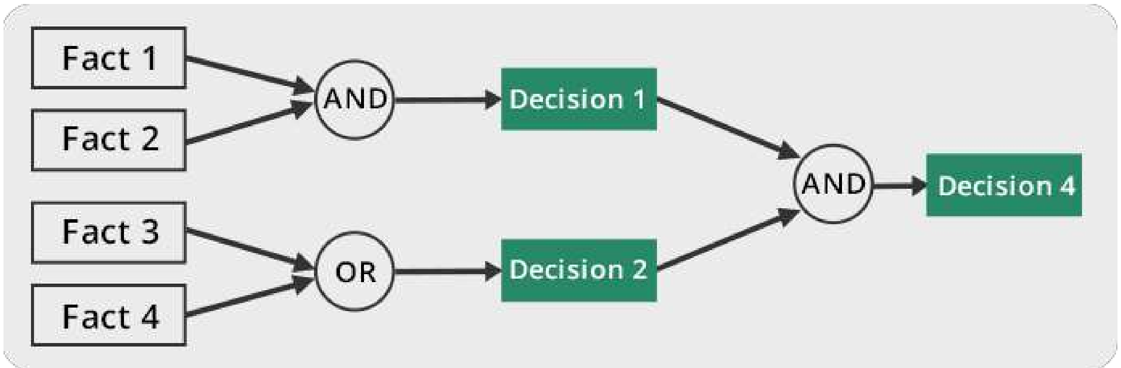
- Forward Chaining
- Backward Chaining

Forward Chaining

It is a strategy of an expert system to answer the question, **"What can happen next?"**

Here, the inferance engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution.

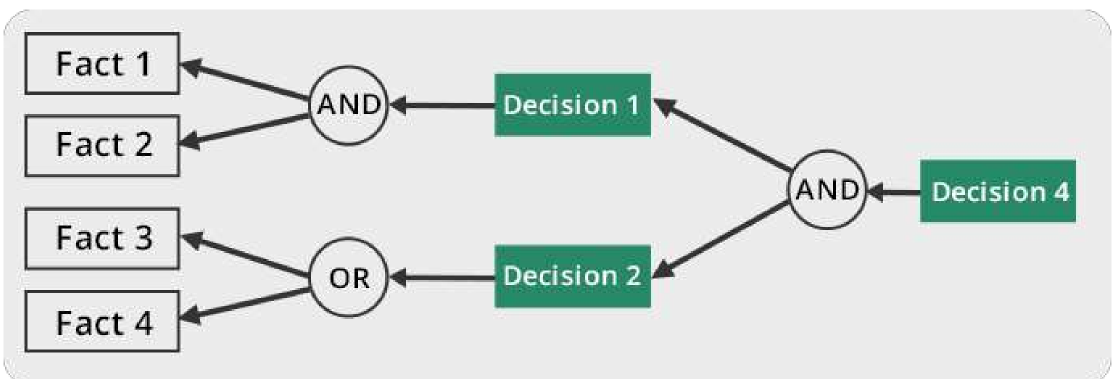
This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.



Backward Chaining

With this strategy, an expert system finds out the answer to the question, **"Why this happened?"**

On the basis of what has already happened, the inference engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.



User Interface

User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence.

It explains how the ES has arrived at a particular recommendation. The explanation may in the following forms:

- Natural language displayed on screen
- Verbal narrations in natural language

- Listing of rule numbers displayed on the screen.

The user interface makes it easy to trace the credibility of the deductions.

Requirements of Efficient ES User Interface

- It should help users to accomplish their goals in shortest possible ay.
- It should be designed to work for user's existing or desired work practices.
- Its technology should be adaptable to user's requirements; not the other way round.
- It should make efficient use of user input.

Expert Systems Limitations

No technology can offer easy and complete solution. Large systems are costly, require significant development time, and computer resources. ESs have their limitations which include:

- Limitations of the technology
- Difficult knowledge acquisition
- ES are Difficult to maintain
- High Development costs

Applications of Expert System

The following table shows where ES can be applied.

Application	Description
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis Systems to deduce cause of disease from observed data, conduction medical operations on humans.
Monitoring Systems	Comparing data continuously with observed system or with prescribed behavior such as leakage monitoring in long petroleum pipeline.
Process Control Systems	Controlling a physical process based on monitoring.
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

Expert System Technology

There are several levels of ES technologies available. Expert systems technologies include:

1. **Expert System Development Environment:** The ES development environment includes hardware and tools. They are:
    - Workstations, minicomputers, mainframes
    - High level Symbolic Programming Languages such as **LIS**t **P**rogramming (LISP) and **PRO**grammation en **LOG**ique (PROLOG).
    - Large databases

2. **Tools:** They reduce the effort and cost involved in developing an expert system to large extent.
    - Powerful editors and debugging tools with multi-windows.
    - They provide rapid prototyping
    - Have Inbuilt definitions of model, knowledge representation, and inference design.

1. **Shells:** A shell is nothing but an expert system without knowledge base. A shell provides the developers with knowledge acquisition, inference engine, user interface, and explanation facility. For example, few shells are given below:
    - Java Expert System Shell (JESS) that provides fully developed Java API for creating an expert system.
    - Vidwan, a shell developed at the National Centre for Software Technology, Mumbai in 1993. It enables knowledge encoding in the form of IF-THEN rules.

Development of Expert Systems: General Steps

The process of ES development is iterative. Steps in developing the ES include:

1. Identify Problem Domain

- The problem must be suitable for an expert system to solve it.
- Find the experts in task domain for the ES project.
- Establish cost-effectiveness of the system.

2. Design the System

- Identify the ES Technology.
- Know and establish the degree of integration with the other systems and databases.
- Realize how the concepts can represent the domain knowledge best.

3. Develop the Prototype

Form Knowledge Base: The knowledge engineer works to:

- Acquire domain knowledge from the expert.
- Represent it in the form of If-THEN-ELSE rules.

4. Test and Refine the Prototype

- The knowledge engineer uses sample cases to test the prototype for any deficiencies in performance.
- End users test the prototypes of the ES.

5. Develop and Complete the ES

- Test and ensure the interaction of the ES with all elements of its environment, including end databases, and other information systems.
- Document the ES project well.
- Train the user to use ES.

6. Maintain the System

- Keep the knowledge base up-to-date by regular review and update.
- Cater for new interfaces with other information systems, as those systems evolve.

Benefits of Expert Systems

- **Availability:** They are easily available due to mass production of software.
- **Less Production Cost:** Production cost is reasonable. This makes them affordable.
- **Speed:** They offer great speed. They reduce the amount of work an individual puts in.
- **Less Error Rate:** Error rate is low as compared to human errors.
- **Reducing Risk:** They can work in the environment dangerous to humans.
- **Steady response:** They work steadily without getting motional, tensed or fatigued.

9. Robotics

Robotics is a domain in artificial intelligence that deals with the study of creating intelligent and efficient robots.

What are Robots?

Robots are the artificial agents acting in real world environment.

Objective

Robots are aimed at manipulating the objects by perceiving, picking, moving, modifying the physical properties of object, destroying it, or to have an effect thereby freeing manpower from doing repetitive functions without getting bored, distracted, or exhausted.

What is Robotics?

Robotics is a branch of AI, which is composed of Electrical Engineering, Mechanical Engineering, and Computer Science for designing, construction, and application of robots.

Aspects of Robotics

- The robots have **mechanical construction**, form, or shape designed to accomplish a particular task.
- They have **electrical components** which power and control the machinery.
- They contain some level of **computer program** that determines what, when and how a robot does something.

Difference in Robot System and Other AI Program

Here is the difference between the two:

AI Programs	Robots
They usually operate in computer-stimulated worlds.	They operate in real physical world
The input to an AI program is in symbols and rules.	Inputs to robots is analog signal in the form of speech waveform or images
They need general purpose computers to operate on.	They need special hardware with sensors and effectors.

Robot Locomotion

Locomotion is the mechanism that makes a robot capable of moving in its environment. There are various types of locomotions:

- Legged
- Wheeled
- Combination of Legged and Wheeled Locomotion
- Tracked slip/skid

Legged Locomotion

- This type of locomotion consumes more power while demonstrating walk, jump, trot, hop, climb up or down, etc.
- It requires more number of motors to accomplish a movement. It is suited for rough as well as smooth terrain where irregular or too smooth surface makes it consume more power for a wheeled locomotion. It is little difficult to implement because of stability issues.
- It comes with the variety of one, two, four, and six legs. If a robot has multiple legs then leg coordination is necessary for locomotion.

The total number of possible **gaits** (a periodic sequence of lift and release events for each of the total legs) a robot can travel depends upon the number of its legs.

If a robot has klegs, then the number of possible events N = (2k-1)!.

In case of a two-legged robot (k=2), the number of possible events is N = (2k-1)!

= (2*2-1)! = 3! = 6.

Hence there are six possible different events:

1. Lifting the Left leg
2. Releasing the Left leg
3. Lifting the Right leg
4. Releasing the Right leg
5. Lifting both the legs together
6. Releasing both the legs together.

In case of k=6 legs, there are 39916800 possible events. Hence the complexity of robots is directly proportional to the number of legs.

Wheeled Locomotion

It requires fewer number of motors to accomplish a movement. It is little easy to implement as there are less stability issues in case of more number of wheels. It is power efficient as compared to legged locomotion.

- **Standard wheel:** Rotates around the wheel axle and around the contact
- **Castor wheel:** Rotates around the wheel axle and the offset steering joint
- **Swedish 45° and Swedish 90° wheels:** Omni-wheel, rotates around the contact point, around the wheel axle, and around the rollers.
- **Ball or spherical wheel:** Omnidirectional wheel, technically difficult to implement.

Slip/Skid Locomotion

In this type, the vehicles use tracks as in a tank. The robot is steered by moving the tracks with different speeds in the same or opposite direction. It offers stability because of large contact area of track and ground.

Components of a Robot

Robots are constructed with the following:

- **Power Supply:** The robots are powered by batteries, solar power, hydraulic, or pneumatic power sources.
- **Actuators**: They convert energy into movement.
- **Electric motors (AC/DC):** They are required for rotational movement.
- **Pneumatic Air Muscles**: They contract almost 40% when air is sucked in them.
- **Muscle Wires**: They contract by 5% when electric current is passed through them.
- **Piezo Motors and Ultrasonic Motors**: Best for industrial robots.
- **Sensors**: They provide knowledge of real time information on the task environment. Robots are equipped with vision sensors to be to compute the depth in the environment. A tactile sensor imitates the mechanical properties of touch receptors of human fingertips.

Computer Vision

This is a technology of AI with which the robots can see. The computer vision plays vital role in the domains of safety, security, health, access, and entertainment.

Computer vision automatically extracts, analyzes, and comprehends useful information from a single image or an array of images. This process involves development of algorithms to accomplish automatic visual comprehension.

Hardware of Computer Vision System

This involves:

- Power supply
- Image acquisition device such as camera
- a processor
- a software
- A display device for monitoring the system
- Accessories such as camera stands, cables, and connectors

Tasks of Computer Vision

OCR: In the domain of computers, Optical Character Reader, a software to convert scanned documents into editable text, which accompanies a scanner.

Face Detection: Many state-of-the-art cameras come with this feature, which enables to read the face and take the picture of that perfect expression. It is used to let a user access the software on correct match.

Object Recognition: They are installed in supermarkets, cameras**,** high-end cars such as BMW, GM, and Volvo.

Estimating Position: It is estimating position of an object with respect to camera as in position of tumor in human's body.

Application Domains of Computer Vision

- agriculture
- autonomous vehicles
- biometrics
- character recognition
- forensics, security, and surveillance
- industrial quality inspection
- face recognition
- gesture analysis
- geoscience
- medical imagery
- pollution monitoring
- process control
- remote sensing
- robotics
- transport

Applications of Robotics

The robotics has been instrumental in the various domains such as:

- **Industries:** Robots are used for handling material, cutting, welding, color coating, drilling, polishing, etc.
- **Military:** Autonomous robots can reach inaccessible and hazardous zones during war. A robot named Daksh, developed by Defense Research and Development Organization (DRDO), is in function to destroy life-threatening objects safely.
- **Medicine:** The robots are capable of carrying out hundreds of clinical tests simultaneously, rehabilitating permanently disabled people, and performing complex surgeries such as brain tumors.

- **Exploration:** The robot rock climbers used for space exploration, underwater drones used for ocean exploration are to name a few.
- **Entertainment:** Disney’s engineers have created hundreds of robots for movie making.

10Neural Networks

Yet another research area in AI, neural networks, is inspired from the natural neural network of human nervous system.

What are Artificial Neural Networks (ANNs)?

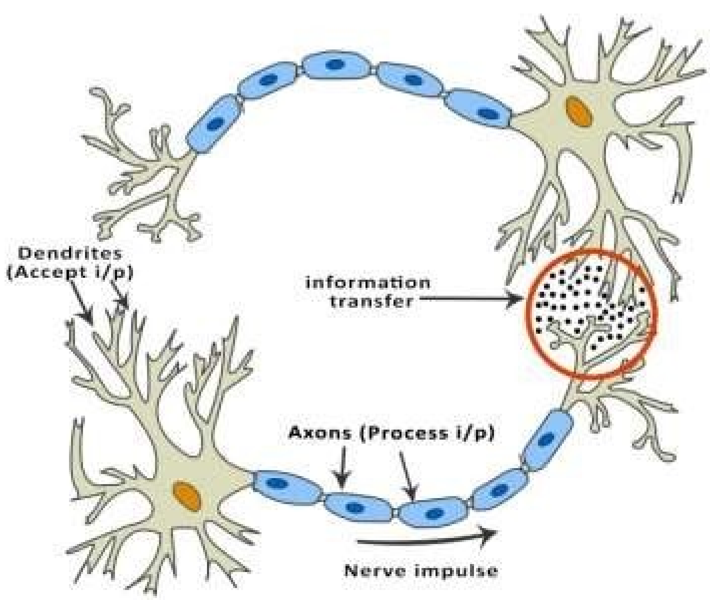
The inventor of the first neurocomputer, Dr. Robert Hecht-Nielsen, defines a neural network as:

"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.”

Basic Structure of ANNs

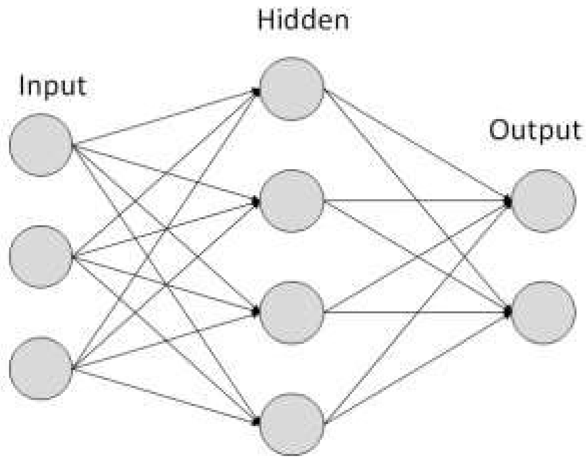
The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living **neurons** and **dendrites**.

The human brain is composed of 100 billion nerve cells called **neurons**. They are connected to other thousand cells by **Axons**. Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward.



ANNs are composed of multiple **nodes,** which imitate biological **neurons** of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its **activation** or **node value**.

Each link is associated with **weight**. ANNs are capable of learning, which takes place by altering weight values. The following illustration shows a simple ANN:

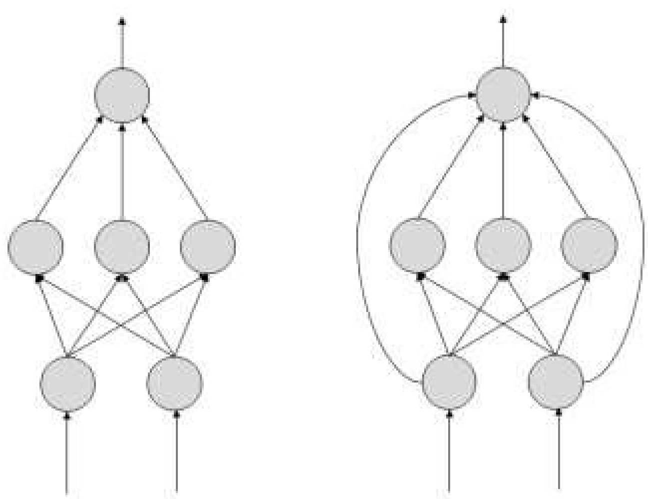


Types of Artificial Neural Networks

There are two Artificial Neural Network topologies: **FeedForward** and **Feedback**.

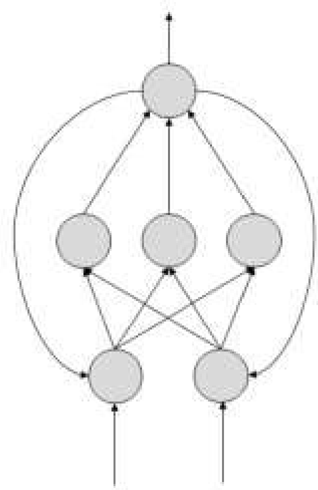
FeedForward ANN

In this ANN, the information flow is unidirectional. A unit sends information to other unit from which it does not receive any information. There are no feedback loops. They are used in pattern generation/recognition/classification. They have fixed inputs and outputs.



Feedback ANN

Here, feedback loops are allowed. They are used in content addressable memories.



Working of ANNs

In the topology diagrams shown, each arrow represents a connection between two neurons and indicates the pathway for the flow of information. Each connection has a weight, an integer number that controls the signal between the two neurons.

If the network generates a “good or desired” output, there is no need to adjust the weights. However, if the network generates a “poor or undesired” output or an error, then the system alters the weights in order to improve subsequent results.

Machine Learning in ANNs

ANNs are capable of learning and they need to be trained. There are several learning strategies:

- **Supervised Learning:** It involves a teacher that is scholar than the ANN itself. For example, the teacher feeds some example data about which the teacher already knows the answers.
  For example, pattern recognizing. The ANN comes up with guesses while recognizing. Then the teacher provides the ANN with the answers. The network then compares it guesses with the teacher’s “correct” answers and makes adjustments according to errors.
- **Unsupervised Learning:** It is required when there is no example data set with known answers. For example, searching for a hidden pattern. In this case, clustering i.e. dividing a set of elements into groups according to some unknown pattern is carried out based on the existing data sets present.

- **Reinforcement Learning:** This strategy built on observation. The ANN makes a decision by observing its environment. If the observation is negative, the network adjusts its weights to be able to make a different required decision the next time.

Back Propagation Algorithm

It is the training or learning algorithm. It learns by example. If you submit to the algorithm the example of what you want the network to do, it changes the network's weights so that it can produce desired output for a particular input on finishing the training.

Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks.

Bayesian Networks (BN)

These are the graphical structures used to represent the probabilistic relationship among a set of random variables. Bayesian networks are also called **Belief Networks** or **Bayes Nets**. BNs reason about uncertain domain.

In these networks, each node represents a random variable with specific propositions. For example, in a medical diagnosis domain, the node Cancerrepresents the proposition that a patient has cancer.

The edges connecting the nodes represent probabilistic dependencies among those random variables. If out of two nodes, one is affecting the other then they must be directly connected in the directions of the effect. The strength of the relationship between variables is quantified by the probability associated with each node.

There is an only constraint on the arcs in a BN that you cannot return to a node simply by following directed arcs. Hence the BNs are called Directed Acyclic Graphs (DAGs).

BNs are capable of handling multivalued variables simultaneously. The BN variables are composed of two dimensions:

1. Range of prepositions
2. Probability assigned to each of the prepositions.

Consider a finite set $X = \{X_1, X_2, \ldots, X_n\}$ of discrete random variables, where each variable X_i may take values from a finite set, denoted by $Val(X_i)$. If there is a directed link from variable X_i to variable, X_j, then variable X_i will be a parent of variable X_j showing direct dependencies between the variables.

The structure of BN is ideal for combining prior knowledge and observed data. BN can be used to learn the causal relationships and understand various problem domains and to predict future events, even in case of missing data.

Building a Bayesian Network

A knowledge engineer can build a Bayesian network. There are a number of steps the knowledge engineer needs to take while building it.

Example problem: Lung cancer A patient has been suffering from breathlessness. He visits the doctor, suspecting he has lung cancer. The doctor knows that barring lung cancer, there are various other possible diseases the patient might have such as tuberculosis and bronchitis.

Gather Relevant Information of Problem

- Is the patient a smoker? If yes, then high chances of cancer and bronchitis.
- Is the patient exposed to air pollution? If yes, what sort of air pollution?
- Take an X-Ray positive X-ray would indicate either TB or lung cancer.

Identify Interesting Variables

The knowledge engineer tries to answer the questions:

- Which nodes to represent?
- What values can they take? In which state can they be?

For now let us consider nodes, with only discrete values. The variable must take on exactly one of these values at a time.

Common types of discrete nodes are:

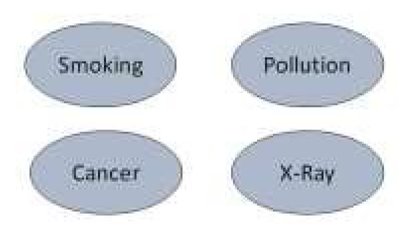
• **Boolean nodes:** They represent propositions, taking binary values TRUE (T) and FALSE (F).

• **Ordered values:** A node Pollution might represent and take values from {low, medium, high} describing degree of a patient's exposure to pollution.

• **Integral values**: A node called Age might represent patient's age with possible values from 1 to 120. Even at this early stage, modeling choices are being made.

Possible nodes and values for the lung cancer example:

Node Name	Type	Value
Pollution	Binary	{LOW, HIGH, MEDIUM}
Smoker	Boolean	{TRUE, FASLE}
Lung-Cancer	Boolean	{TRUE, FASLE}
X-Ray	Binary	{Positive, Negative}

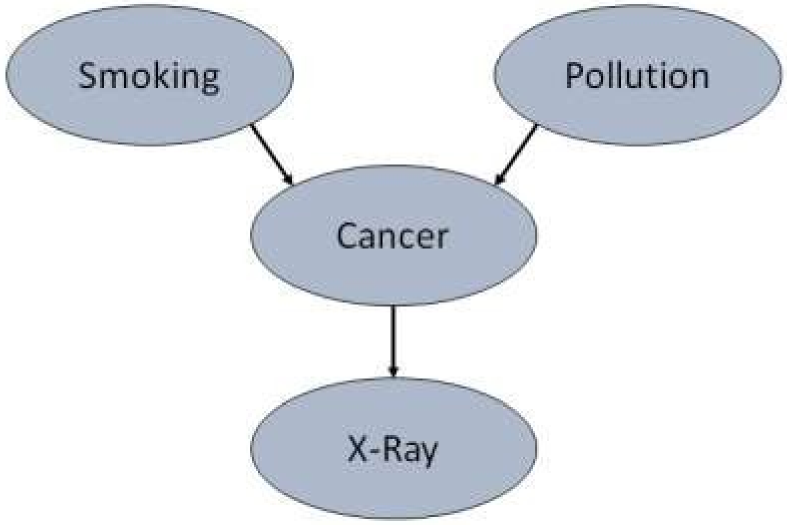


Create Arcs between Nodes

Topology of the network should capture qualitative relationships between variables.

For example, what causes a patient to have lung cancer? - Pollution and smoking. Then add arcs from node Pollution and node Smoker to node Lung-Cancer

Similarly if patient has lung cancer, then X-ray result will be positive. Then add arcs from Lung-Cancer to X-Ray.



Specify Topology

Conventionally, BNs are laid out so that the arcs point from top to bottom. The set of parent nodes of a node X is given by Parents(X).

The Lung-Cancer node has two parents (reasons or causes): Pollution and Smoker, while node Smoker is an **ancestor** of node X-Ray. Similarly, X-Ray is a child (consequence or effects) of node Lung-Cancer and **successor** of nodes Smoker and Pollution.

Conditional Probabilities

Now quantify the relationships between connected nodes: this is done by specifying a conditional probability distribution for each node. As only discrete variables are considered here, this takes the form of a **Conditional Probability Table (CPT)**.

First, for each node we need to look at all the possible combinations of values of those parent nodes. Each such combination is called an **instantiation** of the parent set. For each distinct instantiation of parent node values, we need to specify the probability that the child will take.

For example, the Lung-Cancer node's parents are Pollution and Smoking. They take the possible values = { (H,T), (H,F), (L,T), (L,F)}. The CPT specifies the probability of cancer for each of these cases as <0.05, 0.02, 0.03, 0.001> respectively.

Each node will have conditional probability associated as follows:

Smoking
P(S = T)
0.30

Pollution
P(P = L)
0.90

Lung-Cancer		
P	**S**	**P (C = T|P, S)**
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

X-Ray	
C	**X = (Pos | C)**
T	0.90
F	0.20

Applications of Neural Networks

They can perform tasks that are easy for a human but difficult for a machine:

- **Aerospace**: Autopilot aircrafts, aircraft fault detection.
- **Automotive:** Automobile guidance systems.
- **Military**: Weapon steering, target tracking, object discrimination, facial recognition, signal/image identification.
- **Electronics:** Code sequence prediction, IC chip layout, chip failure analysis, machine vision, voice synthesis.
- **Financial:** Real estate appraisal, loan advisor, mortgage screening, corporate bond rating, portfolio trading program, corporate financial analysis, currency value prediction, document readers, credit application evaluators.
- **Industrial:** Manufacturing process control, product design and analysis, quality inspection systems, welding quality analysis, paper quality prediction, chemical product design analysis, dynamic modeling of chemical process systems, machine maintenance analysis, project bidding, planning, and management.

- **Medical:** Cancer cell analysis, EEG and ECG analysis, prosthetic design, transplant time optimizer.
- **Speech:** Speech recognition, speech classification, text to speech conversion.
- **Telecommunications**: Image and data compression, automated information services, real-time spoken language translation.
- **Transportation:** Truck brake diagnosis, vehicle scheduling, routing systems.
- **Software:** Pattern Recognition in facial recognition, optical character recognition, etc.
- **Time Series Prediction:** ANNs are used to make predictions on stocks and natural calamities.
- **Signal Processing**: Neural networks can be trained to process an audio signal and filter it appropriately in the hearing aids.
- **Control:** ANNs are often used to make steering decisions of physical vehicles.
- **Anomaly Detection:** As ANNs are expert at recognizing patterns, they can also be trained to generate an output when something unusual occurs that misfits the pattern.

11AI Issues

AI is developing with such an incredible speed, sometimes it seems magical. There is an opinion among researchers and developers that AI could grow so immensely strong that it would be difficult for humans to control.

Humans developed AI systems by introducing into them every possible intelligence they could, for which the humans themselves now seem threatened.

Threat to Privacy

An AI program that recognizes speech and understands natural language is theoretically capable of understanding each conversation on e-mails and telephones.

Threat to Human Dignity

AI systems have already started replacing the human beings in few industries. It should not replace people in the sectors where they are holding dignified positions which are pertaining to ethics such as nursing, surgeon, judge, police officer, etc.

Threat to Safety

The self-improving AI systems can become so mighty than humans that could be very difficult to stop from achieving their goals, which may lead to unintended consequences.

12 AI Terminology

Here is the list of frequently used terms in the domain of AI:

Term	Meaning
Agent	Agents are systems or software programs capable of autonomou purposeful and reasoning directed towards one or more goals. They are also called assistants, brokers, bots, droids, intelligent agents, an software agents.
Autonomous Robot	Robot free from external control or influence and able to control itself independently.
Backward Chaining	Strategy of working backward for Reason/Cause of a problem.
Blackboard	It is the memory inside computer, which is used for communicatio between the cooperating expert systems.
Environment	It is the part of real or computational world inhabited by the agent.
Forward Chaining	Strategy of working forward for conclusion/solution of a problem.
Heuristics	It is the knowledge based on Trial-and-error, evaluations, and experimentation.
Knowledge Engineering	Acquiring knowledge from human experts and other resources.
Percepts	It is the format in which the agent obtains information about t environment.
Pruning	Overriding unnecessary and irrelevant considerations in AI systems.
Rule	It is a format of representing knowledge base in Expert System. It is i the form of IF-THEN-ELSE.
Shell	A shell is a software that helps in designing inference engine, knowledge base, and user interface of an expert system.
Task	It is the goal the agent is tries to accomplish.
Turing Test	A test developed by Allan Turing to test the intelligence of a machine as compared to human intelligence.