Expert Systems:

Definition:

An Expert system is a set of program that manipulates encoded knowledge to solve problem in a specialized domain that normally requires human expertise.

A computer system that simulates the **decision- making process** of a human expert in a specific domain.

An expert system's knowledge is obtained from expert sources and coded in a form suitable for the system to use in its inference or reasoning processes. The expert knowledge must be obtained from specialists or other sources of expertise, such as texts, journals, articles and data bases.

An expert system is an "intelligent" program that solves problems in a narrow problem area by using high-quality, specific knowledge rather than an algorithm.

Block Diagram

There is currently no such thing as "standard" expert system. Because a variety of techniques are used to create expert systems, they differ as widely as the programmers who develop them and the problems they are designed to solve. However, the principal components of most expert systems are **knowledge base, an inference engine, and a user interface,** as illustrated in the figure.

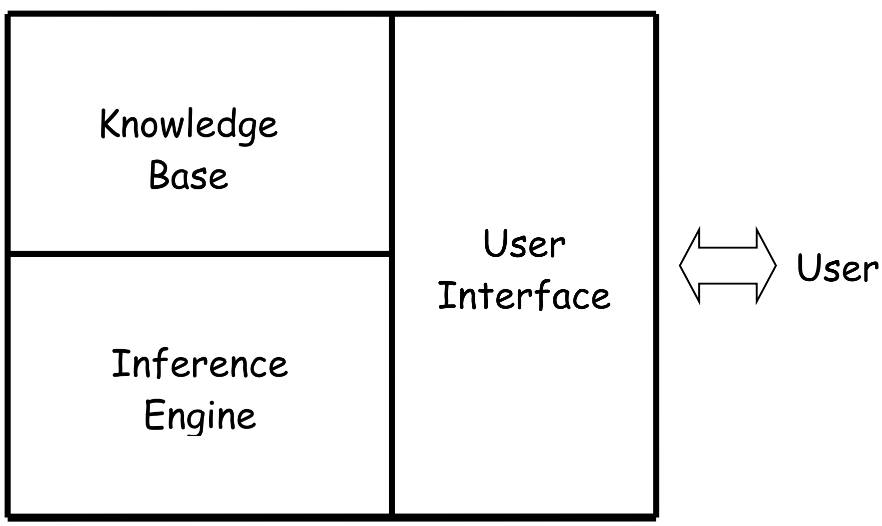


Fig: Block Diagram of expert system

1. **Knowledge Base**

The component of an expert system that contains the system's knowledge is called its knowledge base. This element of the system is so critical to the way most expert systems are constructed that they are also popularly known as *knowledge-based systems*

A knowledge base contains both declarative knowledge (facts about objects, events and situations) and procedural knowledge (information about courses of action). Depending on the form of knowledge representation chosen, the two types of knowledge may be separate or integrated. Although many knowledge representation techniques have been used in expert systems, the most prevalent form of knowledge representation currently used in expert systems is the *rule-based production* system approach.

To improve the performance of an expert system, we should supply the system with some knowledge about the knowledge it posses, or in other words, meta-knowledge.

2. Inference Engine

Simply having access to a great deal of knowledge does not make you an expert; you also must know **how** and **when** to apply the appropriate knowledge. Similarly, just having a knowledge base does not make an expert system intelligent. The system must have another component that directs the implementation of the knowledge. That element of the system is known variously as the *control structure*, the *rule interpreter*, or the *inference engine*.

The inference engine decides which heuristic search techniques are used to determine how the rules in the knowledge base are to be applied to the problem. In effect, an inference engine "runs" an expert system, determining which rules are to be invoked, accessing the appropriate rules in the knowledge base, executing the rules , and determining when an acceptable solution has been found.

3. User Interface

The component of an expert system that communicates with the user is known as the *user interface*. The communication performed by a user interface is bidirectional. At the simplest level, we must be able to describe our problem to the expert system, and the system must be able to respond with its recommendations. We may want to ask the system to explain its "reasoning", or the system may request additional information about the problem from us.

Beside these three components, there is a Working Memory - a data structure which stores information about a specific run. It holds current facts and knowledge.

<u>Stages of Expert System Development:</u>

Although great strides have been made in expediting the process of developing an expert system, it often remains an extremely time consuming task. It may be possible for one or two people to develop a small expert system in a few months; however the development of a sophisticated system may require a team of several people working together for more than a year.

An expert system typically is developed and refined over a period of several years. We can divide the process of expert system development into five distinct stages. In practice, it may not be possible to break down the expert system development cycle precisely. However, an examination of these five stages may serve to provide us with some insight into the ways in which expert systems are developed.

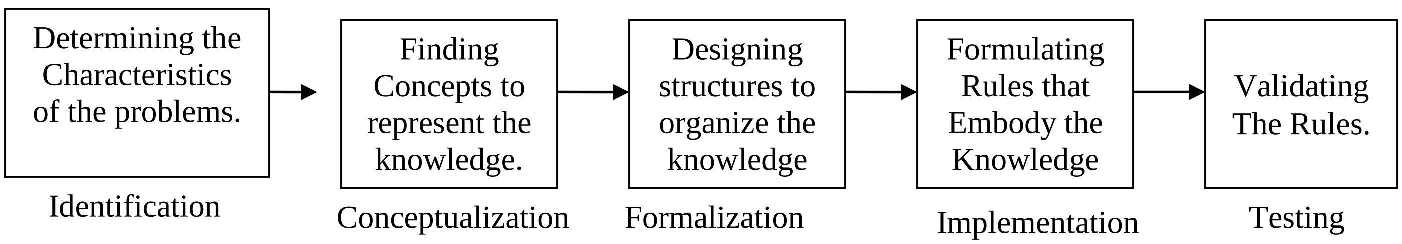


Fig: Different phases of expert system development

Identification:

Beside we can begin to develop an expert system, it is important that we describe, with as much precision as possible, the problem that the system is intended to solve. It is not enough simply to feel that the system would be helpful in certain situation; we must determine the exact nature of the problem and state the precise goals that indicate exactly how we expect the expert system to contribute to the solution.

Conceptualization:

Once we have formally identified the problem that an expert system is to solve, the next stage involves analyzing the problem further to ensure that its specifics, as well as it generalities, are understood. In the conceptualization stage the knowledge engineer frequently creates a diagram of the problem to depict graphically the relationships between the objects and processes in the problem domain. It is often helpful at this stage to divide the problem into a series of sub-problems and to diagram both the relationships among the pieces of each sub-problem and the relationships among the various sub-problems.

Formalization:

In the preceding stages, no effort has been made to relate the domain problem to the artificial intelligence technology that may solve it. During the identification and the conceptualization stages, the focus is entirely on understanding the problem. Now, during the formalization stage,

the problem is connected to its proposed solution, an expert system, by analyzing the relationships depicted in the conceptualization stage.
During formalization, it is important that the knowledge engineer be familiar with the following:

- The various techniques of knowledge representation and heuristic search used in expert systems.
- The expert system "tools" that can greatly expedite the development process. And
- Other expert systems that may solve similar problems and thus may be adequate to the problem at hand.

Implementation:

During the implementation stage, the formalized concepts are programmed onto the computer that has been chosen for system development, using the predetermined techniques and tools to implement a "first pass" prototype of the expert system.

Theoretically, if the methods of the previous stage have been followed with diligence and care, the implementation of the prototype should be as much an art as it is a science, because following all rules does not guarantee that the system will work the first time it is implemented. Many scientists actually consider the first prototype to be a "throw-away' system, useful for evaluating progress but hardly a usable expert system.

Testing:

Testing provides opportunities to identify the weakness in the structure and implementation of the system and to make the appropriate corrections. Depending on the types of problems encountered, the testing procedure may indicate that the system was

Features of an expert system:

What are the features of a good expert system ? Although each expert system has its own particular characteristics, there are several features common to many systems. The following list from Rule-Based Expert Systems suggests seven criteria that are important prerequisites for the acceptance of an expert system .

1. "The program should be **useful**." An expert system should be developed to meet a specific need, one for which it is recognized that assistance is needed.

2. "The program should be **usable**." An expert system should be designed so that even a novice computer user finds it easy to use .

3. "The program should be **educational when appropriate**." An expert system may be used by non-experts, who should be able to increase their own expertise by using the system.

4. “The program should be able to **explain its advice**.” An expert system should be able to explain the “reasoning” process that led it to its conclusions, to allow us to decide whether to accept the system’s recommendations.

5. “The program should be able to **respond to simple questions**.” Because people with different levels of knowledge may use the system, an expert system should be able to answer questions about points that may not be clear to all users.

6. “The program should be able to **learn new knowledge**.” Not only should an expert system be able to respond to our questions, it also should be able to ask questions to gain additional information.

7. “The program’s knowledge should be **easily modified**.” It is important that we should be able to revise the knowledge base of an expert system easily to correct errors or add new information.

Natural Language Processing:

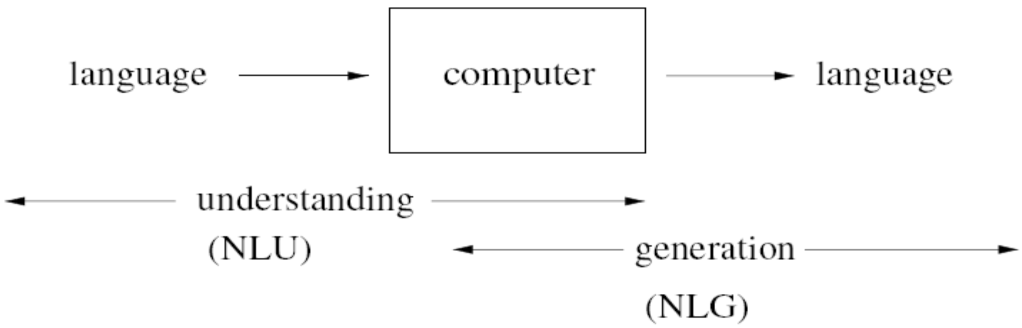
Perception and communication are essential components of intelligent behavior. They provide the ability to effectively interact with our environment. Humans perceive and communicate through their five basic senses of sight, hearing, touch, smell and taste, and their ability to generate meaningful utterances. Developing programs that understand a natural language is a difficult problem. Natural languages are large. They contain infinity of different sentences. No matter how many sentences a person has heard or seen, new ones can always be produced. Also, there is much ambiguity in a natural language. Many words have several meanings and sentences can have different meanings in different contexts. This makes the creation of programs that understand a natural language, one of the most challenging tasks in AI.

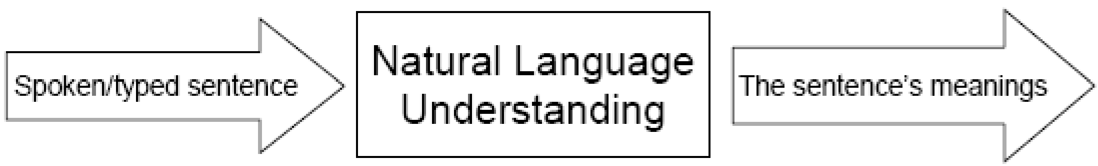
Developing programs to understand natural language is important in AI because a natural form of communication with systems is essential for user acceptance. AI programs must be able to communicate with their human counterparts in a natural way, and natural language is one of the most important mediums for that purpose. So, Natural Language Processing (NLP) is the field that deals with the computer processing of natural languages, mainly evolved by people working in the field of Artificial Intelligence.

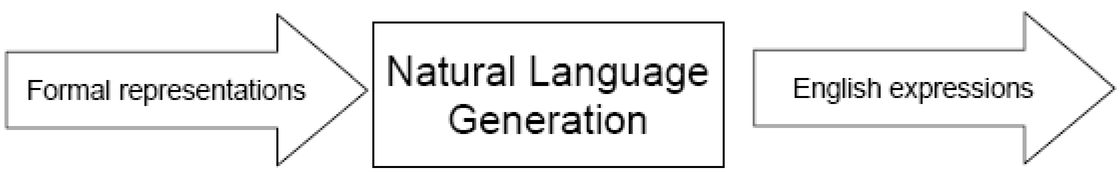
Natural Language Processing (NLP), is the attempt to extract the fuller meaning representation from the free text. Natural language processing is a technology which involves converting spoken or written human language into a form which can be processed by computers, and vice versa. Some of the better-known applications of NLP include:

- **Voice recognition software** which translates speech into input for word processors or other applications;
- **Text-to-speech synthesizers** which read text aloud for users such as the hearing-impaired;
- **Grammar and style checkers** which analyze text in an attempt to highlight errors of grammar or usage;
- **Machine translation systems** which automatically render a document such as a web page in another language.

computers using natural language as input and/or output







Natural Language Generation:

"Natural Language Generation (NLG), also referred to as text generation, is a subfield of natural language processing (NLP; which includes computational linguistics)

Natural Language Generation (NLG) is the natural language processing task of generating natural language from a machine representation system such as a knowledge base or a logical form.

In a sense, one can say that an NLG system is like a translator that converts a computer based representation into a natural language representation. However, the methods to produce the final language are very different from those of a compiler due to the inherent expressivity of natural languages.

NLG may be viewed as the opposite of natural language understanding. The difference can be put this way: whereas in natural language understanding the system needs to disambiguate the input sentence to produce the machine representation language, in NLG the system needs to make decisions about how to put a concept into words.

The different types of generation techniques can be classified into four main categories:

- Canned text systems constitute the simplest approach for single-sentence and multi-sentence text generation. They are trivial to create, but very inflexible.
- Template systems, the next level of sophistication, rely on the application of pre-defined templates or schemas and are able to support flexible alterations. The template approach is used mainly for multi-sentence generation, particularly in applications whose texts are fairly regular in structure.
- Phrase-based systems employ what can be seen as generalized templates. In such systems, a phrasal pattern is first selected to match the top level of the input, and then each part of the pattern is recursively expanded into a more specific phrasal pattern that matches some subportion of the input. At the sentence level, the phrases resemble phrase structure grammar rules and at the discourse level they play the role of text plans.
- Feature-based systems, which are as yet restricted to single-sentence generation, represent each possible minimal alternative of expression by a single feature. Accordingly, each sentence is specified by a unique set of features. In this framework, generation consists in the incremental collection of features appropriate for each portion of the input. Feature collection itself can either be based on unification or on the traversal of a feature selection network. The expressive power of the approach is very high since any distinction in language can be added to the system as a feature. Sophisticated feature-based generators, however, require very complex input and make it difficult to maintain feature interrelationships and control feature selection.

Many natural language generation systems follow a hybrid approach by combining components that utilize different techniques.

Natural Language Understanding:

Developing programs that understand a natural language is a difficult problem. Natural languages are large. They contain infinity of different sentences. No matter how many sentences a person has heard or seen, new ones can always be produced. Also, there is much ambiguity in a natural language. Many words have several meaning such as can, bear, fly, bank etc, and sentences have different meanings in different contexts.

Example :- a *can* of juice. I *can* do it.

This makes the creation of programs that understand a natural language, one of the most challenging tasks in AI. Understanding the language is not only the transmission of words. It also requires inference about the speakers' goal, knowledge as well as the context of the interaction. We say a program understand natural language if it behaves by taking the correct or acceptable action in response to the input. A word functions in a sentence as a part of speech. Parts of the speech for the English language are nouns, pronouns, verbs, adjectives, adverbs, prepositions, conjunctions and interjections. Three major issues involved in understanding language.

- A large amount of human knowledge is assumed.
- Language is pattern based, phonemes are components of the words and words make phrases and sentences. Phonemes, words and sentences order are not random.
- Language acts are the product of agents (human or machine).

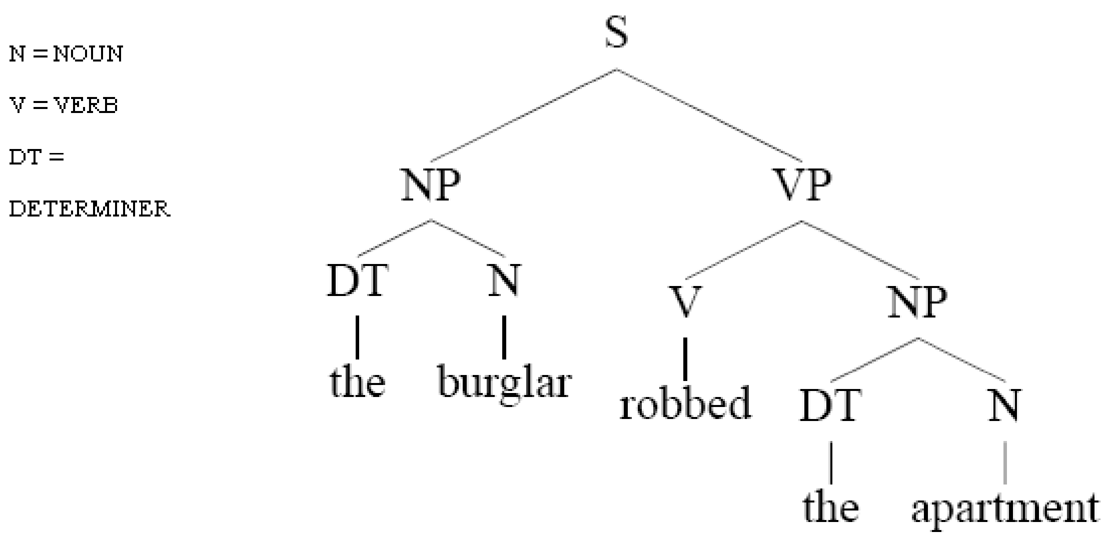
Levels of knowledge used in Language Understanding

A language understanding knowledge must have considerable knowledge about the structures of the language including what the words are and how they combine into phrases and sentences. It must also know the meanings of the words and how they contribute to the meanings of the sentence and to the context within which they are being used. The component forms of knowledge needed for an understanding of natural languages are sometimes classified according to the following levels.

- **Phonological**
    - Relates sound to the words we recognize. A phoneme is the smallest unit of the sound. Phones are aggregated to the words.
- **Morphological**
    - This is lexical knowledge which relates to the word construction from basic units called morphemes. A morpheme is the smallest unit of meaning. Eg:- *friend* + ly = friendly
- **Syntactic**
    - This knowledge relates to how words are put together or structure red together to form grammatically correct sentences in the language.
- **Semantic**
    - This knowledge is concerned with the meanings of words and phrases and how they combine to form sentence meaning.
- **Pragmatic**
    - This is high – level knowledge which relates to the use of sentences in different contexts and how the context affects the meaning of the sentence.
- **World**
    - Includes the knowledge of the physical world, the world of human social interaction, and the roles of goals and intentions in communication.

Basic Parsing Techniques

Before the meaning of a sentence can be determined, the meanings of its constituent parts must be established. This requires knowledge of the structure of the sentence, the meaning of the individual words and how the words modify each other. The process of determining the syntactical structure of a sentence is known as parsing. Parsing is the process of analyzing a sentence by taking it apart word – by – word and determining its structure from its constituent parts and sub parts. The structure of a sentence can be represented with a syntactic tree. When given an input string, the lexical parts or terms (root words), must first be identified by type and then the role they play in a sentence must be determined. These parts can be combined successively into larger units until a complete tree has been computed.



Noun Phrases (NP): “the burglar”, “the apartment”

Verb Phrases (VP): “robbed the apartment”

Sentences (S): “the burglar robbed the apartment”

To determine the meaning of a word, a parser must have access to a lexicon. When the parser selects the word from the input stream, it locates the world in the lexicon and obtains the word’s possible functions and features, including the semantic information.

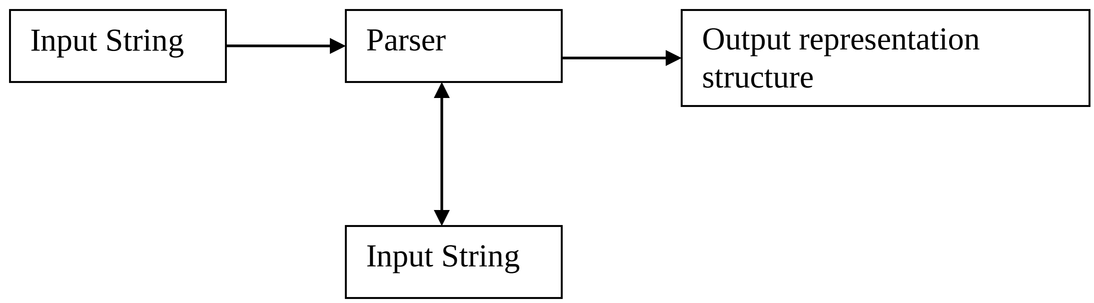


Figure :- Parsing an input to create an output structure

Lexeme (Lexicon) & word forms:

The distinction between these two senses of "word" is arguably the most important one in morphology. The first sense of "word", the one in which *dog* and *dogs* are "the same word", is called a lexeme. The second sense is called *word form*. We thus say that *dog* and *dogs* are different forms of the same lexeme. *Dog* and *dog catcher,* on the other hand, are different lexemes, as they refer to two different kinds of entities. The form of a word that is chosen conventionally to represent the canonical form of a word is called a lemma, or citation form.

A lexicon defines the words of a language that a system knows about. This is includes common words and words that are specific to the domain of the application. Entries include meanings for each word and its syntactic and morphological behavior.

Morphology:

Morphology is the identification, analysis and description of the structure of words (words as units in the lexicon are the subject matter of lexicology). While words are generally accepted as being (with clitics) the smallest units of syntax, it is clear that in most (if not all) languages, words can be related to other words by rules. For example, English speakers recognize that the words *dog*, *dogs*, and *dog catcher* are closely related. English speakers recognize these relations from their tacit knowledge of the rules of word formation in English. They infer intuitively that *dog* is to *dogs* as *cat* is to *cats*; similarly, *dog* is to *dog catcher* as *dish* is to *dishwasher* (in one sense). The rules understood by the speaker reflect specific patterns (or regularities) in the way words are formed from smaller units and how those smaller units interact in speech. In this way, morphology is the branch of linguistics that studies patterns of word formation within and across languages, and attempts to formulate rules that model the knowledge of the speakers of those languages.

Morphological analysis is the process of recognizing the suffixes and prefixes that have been attached to a word.

We do this by having a table of affixes and trying to match the input as:
prefixes +root + suffixes.

- For example: adjective + ly -> adverb. E.g.: [Friend + ly]=friendly
- We may not get a unique result.
- "-s, -es" can be either a plural noun or a 3ps verb
- "-d, -ed" can be either a past tense or a perfect participle

Morphological Information:

- Transform part of speech
  - *green, greenness (adjective, noun)*
  - *walk, walker (verb, noun)*

- Change features of nouns
  - *boat, boats (singular, plural)*

- Bill slept , Bill's bed
  - (subjective case, possessive case)

- Change features of verbs
  - Aspect
    - *I walk. I am walking.* (present, progressive)
  - Tense
    - *I walked. I will walk. I had been walking. (past, future, past progressive)*
  - Number and person
    - *I walk. They walk. (first person singular, third person plural)*

Syntactic Analysis:

Syntactic analysis takes an input sentence and produces a representation of its grammatical structure. A grammar describes the valid parts of speech of a language and how to combine them into phrases. The grammar of English is nearly context free.

A computer grammar specifies which sentences are in a language and their parse trees. A parse tree is a hierarchical structure that shows how the grammar applies to the input. Each level of the tree corresponds to the application of one grammar rule.

It is the starting point for working out the meaning of the whole sentence. Consider the following two sentences.

1. "The dog ate the bone."
2. "The bone was eaten by the dog."

Understanding the structure (via the syntax rules) of the sentences help us work out that it's the bone that gets eaten and not the dog. Syntactic analysis determines possible grouping of words in a sentence. In other cases there may be many possible groupings of words. Consider the sentence "John saw Mary with a telescope". Two different readings based on the groupings.

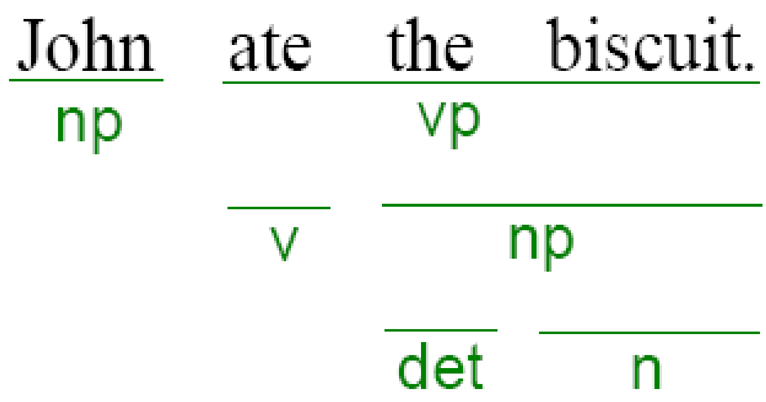
1. John saw (Mary with a telescope).
2. John (saw Mary with a telescope).

A sentence is syntactically ambiguous if there are two or more possible groupings. Syntactic analysis helps determining the meaning of a sentence by working out possible word structure. Rules of syntax are specified by writing a *grammar* for the language. A parser will check if a sentence is correct according to the grammar. It returns a representation (parse tree) of the sentence's structure. A grammar specifies allowable sentence structures in terms of basic categories such as noun and verbs. A given grammar, however, is unlikely to cover all possible grammatical sentences. Parsing sentences is to help determining their meanings, not just to check that they are correct. Suppose we want a grammar that recognizes sentences like the following.

John ate the biscuit.
The lion ate the zebra.
The lion kissed John

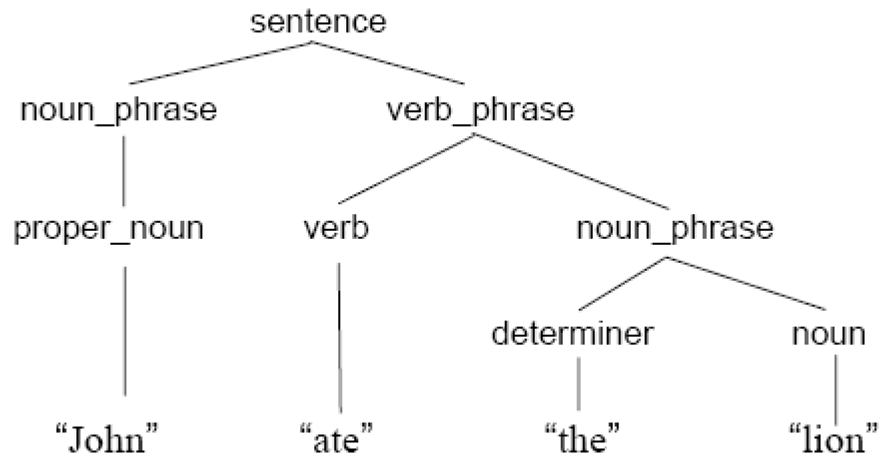
But reject incorrect sentences such as
Ate John biscuit the.
Zebra the lion the ate.
Biscuit lion kissed.

A simple grammar that deals with this is given below
sentence --> noun_phase, verb phrase.
noun_phrase --> proper_noun.
noun_phrase --> determiner, noun.
verb_phrase --> verb, noun_phrase.
proper_noun --> [mary].
proper_noun --> [john].
noun --> [zebra].
noun --> [biscuit].
verb --> [ate].
verb --> [kissed].
determiner --> [the].



Incorrect sentences like “biscuit lion kissed” will be excluded by the grammar.

- A ***parse trees*** illustrates the syntactic structure of the sentence.



Semantic Analysis:

Semantic analysis is a process of converting the syntactic representations into a meaning representation.

This involves the following tasks:

- Word sense determination
- Sentence level analysis
- Knowledge representation

- Word sense

Words have different meanings in different contexts.

Mary had a bat in her office.

- bat = `a baseball thing’

- bat = `a flying mammal'

- Sentence level analysis

Once the words are understood, the sentence must be assigned some meaning

I saw an astronomer with a telescope.

- Knowledge Representation

Understanding language requires lots of knowledge**.**

- Using predicate logic, for example, one can represent sentences like

"John likes Mary" $\quad$ `likes(john,mary)`

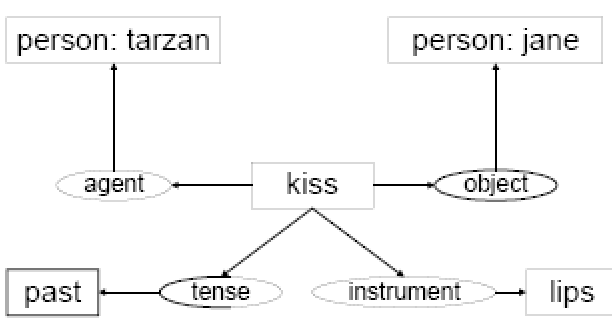
"The man likes Mary" $\quad$ $man(m1) \wedge likes(m1, mary)$

"A man likes Mary" $\quad$ $\exists X (man(X) \wedge likes(X, mary))$

"A tall bearded man likes Mary"

$$\exists X (man(X) \wedge tall(X) \wedge bearded(X) \wedge likes(X, mary))$$

- Using semantic net, one can represent sentence "Tarzan kissed Jane" as



Parameters in Natural Language Processing:

- Auditory Inputs
- Segmentation
- Syntax Structure
- Semantic Structure
- Pragmatic Analysis

- Auditory Inputs:

Three of our five senses – sight, hearing and touch – are used as *major inputs*. These are usually referred to as the visual, auditory and tactile inputs respectively. They are sometimes called input channels; however, as previously mentioned, the term "channel" is used in various ways, so I will avoid it.

In the fashion of video devices, audio devices are used to either capture or create sound. In some cases, an audio output device can be used as an input device, in order to capture produced sound.

- Microphone
- MIDI keyboard or other digital musical instrument

- Segmentation:

Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. The term applies both to mental processes used by humans when reading text, and to artificial processes implemented in computers, which are the subject of natural language processing. The problem is non-trivial, because while some written languages have explicit word boundary markers, such as the word spaces of written English and the distinctive initial, medial and final letter shapes of Arabic, such signals are sometimes ambiguous and not present in all written languages.

Word segmentation is the problem of dividing a string of written language into its component words. In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word delimiter. (Some examples where the space character alone may not be sufficient include contractions like *can't* for *can not*.)

However the equivalent to this character is not found in all written scripts, and without it word segmentation is a difficult problem. Languages which do not have a trivial word segmentation process include Chinese, Japanese, where sentences but not words are delimited, and Thai, where phrases and sentences but not words are delimited.

In some writing systems however, such as the Ge'ez script used for Amharic and Tigrinya among other languages, words are explicitly delimited (at least historically) with a non-whitespace character.

Word splitting is the process of parsing concatenated text (i.e. text that contains no spaces or other word separators) to infer where word breaks exist.

Sentence segmentation is the problem of dividing a string of written language into its component sentences. In English and some other languages, using punctuation, particularly the full stop character is a reasonable approximation. However, even in English this problem is not trivial due to the use of the full stop character for abbreviations, which may or may not also terminate a sentence. For example *Mr.* is not its own sentence in "*Mr. Smith went to the shops in Jones Street."* When processing plain text, tables of abbreviations that contain periods can help prevent incorrect assignment of sentence boundaries. As with word segmentation, not all written languages contain punctuation characters which are useful for approximating sentence boundaries.

Other segmentation problems: Processes may be required to segment text into segments besides words, including morphemes (a task usually called morphological analysis), paragraphs, topics or discourse turns.

A document may contain multiple topics, and the task of computerized text segmentation may be to discover these topics automatically and segment the text accordingly. The topic boundaries may be apparent from section titles and paragraphs. In other cases one needs to use techniques similar to those used in document classification. Many different approaches have been tried.

- Syntax Structure:

Same concept as in the syntactic analysis above

- Semantic Structure:

Same concept as in the semantic analysis above

- Pragmatic Analysis:

This is high level knowledge which relates to the use of sentences in different contexts and how the context affects the meaning of the sentences. It is the study of the ways in which language is used and its effect on the listener. Pragmatic comprises aspects of meaning that depend upon the context or upon facts about real world.

Pragmatics – Handling Pronouns

Handling pronouns such as “he”, “she” and “it” is not always straight forward. Let us see the following paragraph.

“John buys a new telescope. He sees Mary in the distance. He gets out his telescope. He looks at her through it”.

Here, “*her*” refers to *Mary* who was not mentioned at all in the previous sentences. John’s telescope was referred to as “*a new telescope*”, “*his telescope*” and “*it*”.

Let us see one more example

> "When is the next flight to Sydney?"
> "Does it have any seat left?"

Here, "*it*", refers to a particular flight to Sydney, not Sydney itself.

Pragmatics – Ambiguity in Language

A sentence may have more than one structure such as

"I saw an astronomer with a telescope."

This English sentence has a prepositional phrase "*with a telescope*" which may be attached with either with verb to make phrase "*saw something with telescope*" or to object noun phrase to make phrase "*a astronomer with a telescope*". If we do first, then it can be interpreted as "*I saw an astronomer who is having a telescope*", and if we do second, it can be interpreted as "*Using a telescope I saw an astronomer*".

Now, to remove such ambiguity, one possible idea is that we have to consider the context. If the knowledge base (KB) can prove that whether the telescope is with astronomer or not, then the problem is solved.

Next approach is that; let us consider the real scenario where the human beings communicate. If A says the same sentence "*I saw an astronomer with a telescope*." To B, then in practical, it is more probable that, B (listener) realizes that "*A has seen astronomer who is having a telescope*". It is because, normally, the word "*telescope*" belongs to "*astronomer*", so it is obvious that B realizes so.

If A has says that "*I saw a lady with a telescope*." In this case, B realizes that "*A has seen the lady using a telescope*", because the word "*telescope*" has not any practical relationship with "*lady*" like "*astronomer*".

So, we may be able to remove such ambiguity, by defining a data structure, which can efficiently handle such scenario. This idea may not 100% correct but seemed more probable.

What is Natural language Processing?

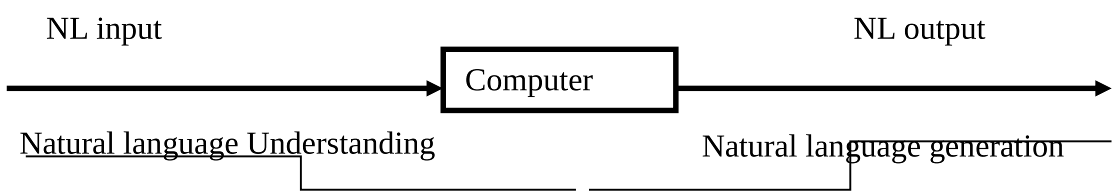
Natural Language Processing (NLP) aims to acquire, understand and generate the human languages such as English, French, Tamil, Hindi, etc. A language is a system, a set of symbols and a set of rules (or grammar).

NLP is a convenient description for all attempts to use computers to process natural language. NLP is also an area of artificial intelligence research that attempts to reproduce the human interpretation of language for computer system processing. The ultimate goal of NLP is to determine a system of language, words, relations, and conceptual information that can be used by computer logic to implement artificial language interpretation.

NLP includes anything a computer needs to understand natural language (written or spoken) and also generate the natural language. To build computational natural language systems, we need Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLG systems convert information from computer databases into normal-sounding human language, and NLU systems convert samples of human language into more representation that are easier for computer programs to manipulate.

Components of NLP

NLP encompasses anything a computer needs to understand natural language (typed or spoken) and also generate the natural language.



Natural language understanding (NLU)

The NLU task is understanding and reasoning while the input is a natural language. Here we ignore the issues of natural language generation.

Natural Language generation (NLG)

NLG is a subfield of natural language processing NLP.NLG is also referred to text generation.

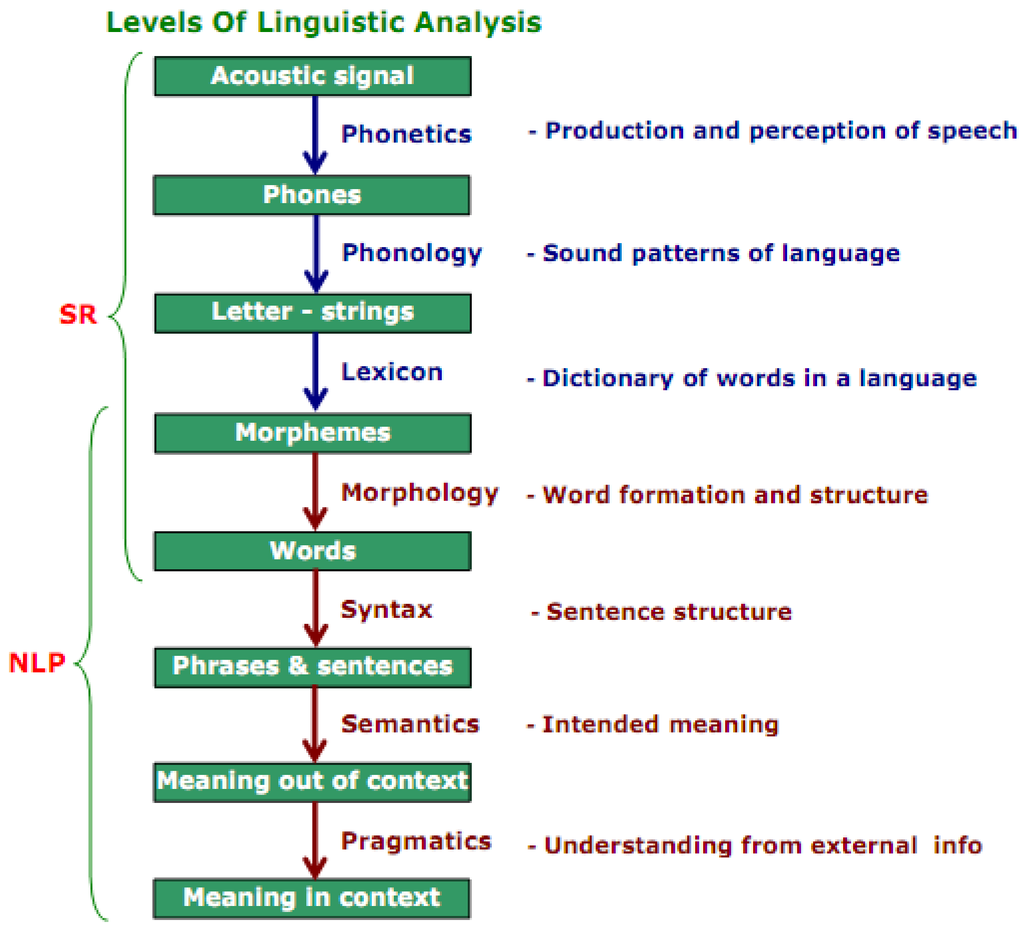
Major Application of Natural Language Processing

NLP is having a very important place in our day-to-day life due to its large natural language applications. By means of these NLP applications the user can interact with computers in their own mother tongue by means of a keyword and a screen. The few NLP processes are:

- Part-of-speech tagging
- Information retrieval
- Machine translation
- Question answering
- Spoken dialogue system
- Speech recognition etc.

Steps of Natural Language Processing (NLP)

Natural Language Processing is done at different levels. These levels are briefly stated below.



The levels of linguistic analysis can be categorized in to two levels. Higher level corresponds to Speech Recognition (SR) and lower levels correspond to Natural Language Processing (NLP).

Phonological Analysis: Phonology is the study of sound system in a language. The minimal unit of sound system is the phoneme which is capable of distinguishing the meanings in the words. The phonemes combine to form a higher level unit called syllable and syllables combine to form the words. Therefore, the organization of the sounds in a language exhibits the linguistic as well as computational challenges for its analysis.

Morphological Analysis: This level deals with the componential nature of words, which are composed of morphemes – the smallest units of semantic meaning. For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix pre, the root 'registra', and the suffix '-tion'. Since the meaning of each morpheme remains the same across words, humans can break down an unknown word into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffix '-ed' to a verb, conveys that the action of the verb took place in the past. This is a key piece of meaning, and in fact, is frequently only evidenced in a text by the use of the -ed morpheme.

Lexical Analysis: At this level, humans, as well as NLP systems, interpret the meaning of individual words. Several types of processing contribute to word-level understanding – the first of these being assignment of a single part-of-speech (POS) tag to each word. In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of speech tag based on the context in which they occur. The lexical level may require a lexicon, and the particular approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon.

Syntactic Analysis: Syntactic analysis uses the results of morphological analysis and lexical analysis to build a structural description of the sentence. The goal of this process, called parsing, is to convert the flat list of words that forms the sentence into a structure that defines the units that are represented by that flat list. The important thing here is that a flat list of words has been converted into a hierarchical structure and that the structures correspond to meaning units when semantic analysis is performed.

Semantic Analysis: It derives an absolute (dictionary definition) meaning from context; it determines the possible meaning of a sentence in a context .The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between individual words into appropriate objects in the knowledge base or data base. It must create the correct structure s to correspond to the way the meaning of the individual words combine with each other. The structures for which no such mapping is possible are rejected.

Example: the sentence "colorless green ideas……..” would be rejected as it has no such semantic mapping, because colorless and green make no sense.

Discourse Integration: The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it.

Example: the meaning of word "it" in the sentence, "you wanted it" depends on the previous discourse context.

Pragmatic Analysis: It derives knowledge from external commonsense information; it means understanding the purposeful use of language in situations, particularly those aspects of language which require world knowledge.

Example: If someone says “the door is open” then it is necessary to know which door “the door” refers to; here it is necessary to know what the intention of the speaker: could be a pure statement of fact, could be an explanation of how the cat got in, or could be a request to the person addressed to close the door.

Note:

- If sentence has more than one interpretation (parse tree), such sentence is called ambiguous sentence.
- The grammar which generates more than one parse tree of a sentence is called ambiguous grammar.
- Parsing/ syntax analysis is the mechanism of identifying whether a given sentence is grammatically correct or not.
- Semantic analysis is the mechanism of extracting meanings of the sentence.
- Pragmatic analysis is used to extract the contextual/ situational meaning of a sentence. It is used when a sentence has more than one semantic interpretation.
- Parse tree is a tree that is used in syntax analysis of parsing of a sentence.

Machine Vision

•The goal of Machine Vision is to create a model of the real world from images.

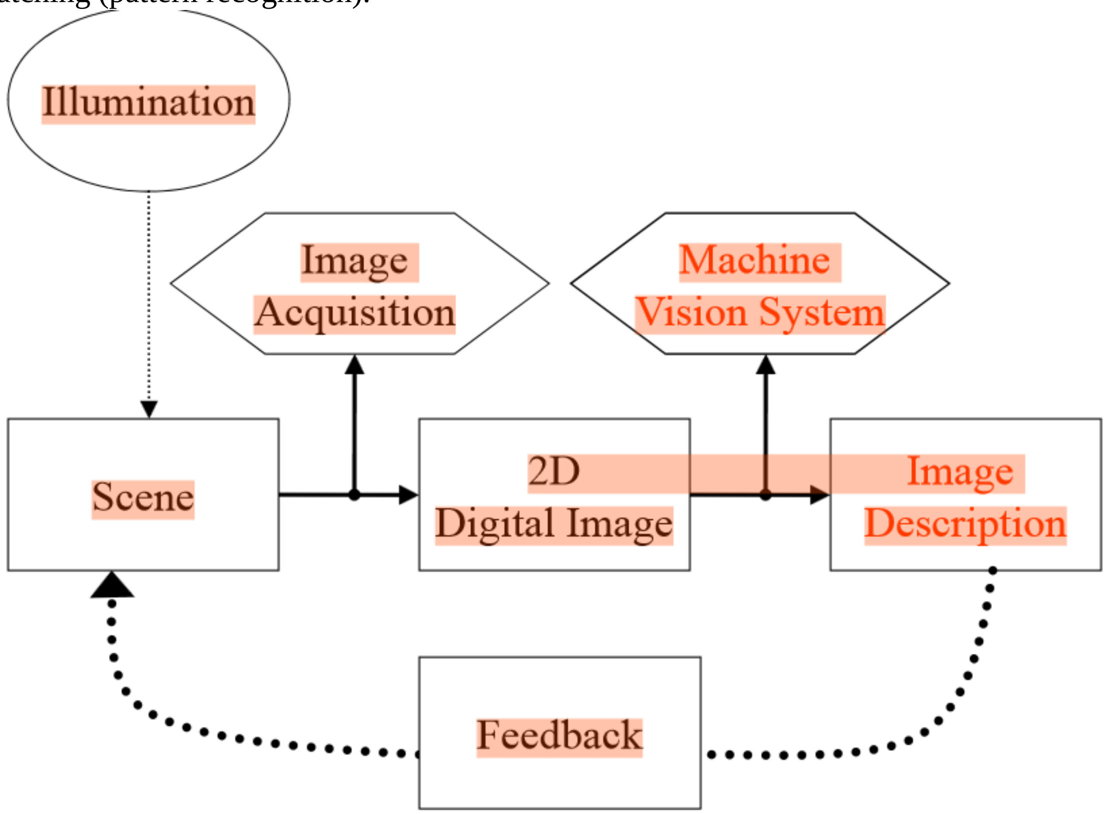
–A machine vision system recovers useful information about a scene from its two dimensional projections.

–The world is three dimensional

–Two dimensional digitized images

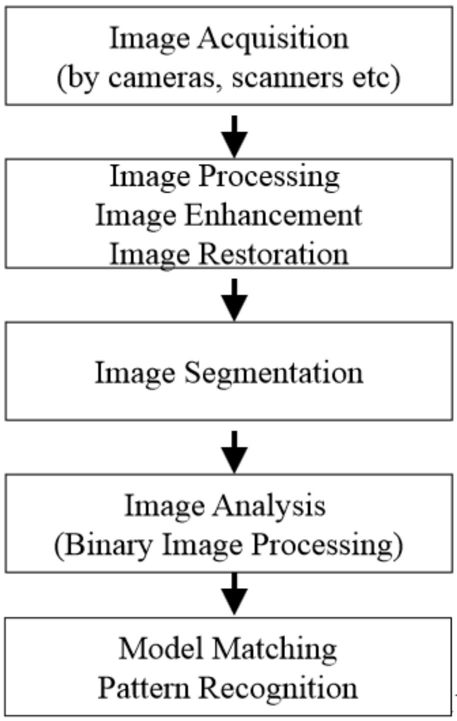
•Knowledge about the objects (regions) in a scene and projection geometry is required.

•The information which is recovered differs depending on the application –Satellite, medical images etc. •Processing takes place in stages: –Enhancement, segmentation, image analysis and matching (pattern recognition).



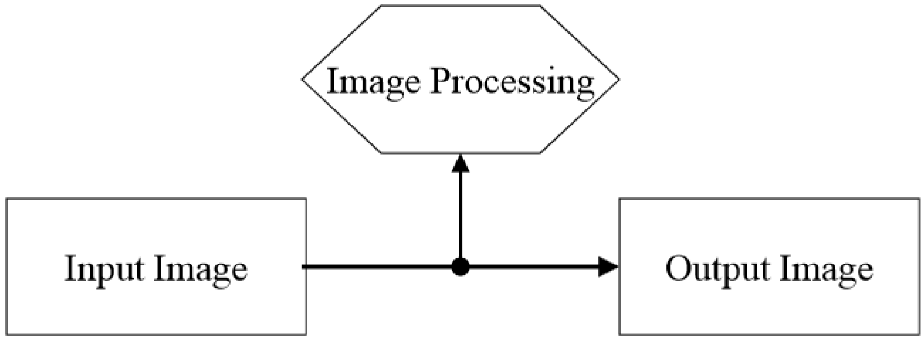
The goal of a machine vision system is to compute a meaningful description of the scene (e.g., object)

Machine Vision Stages



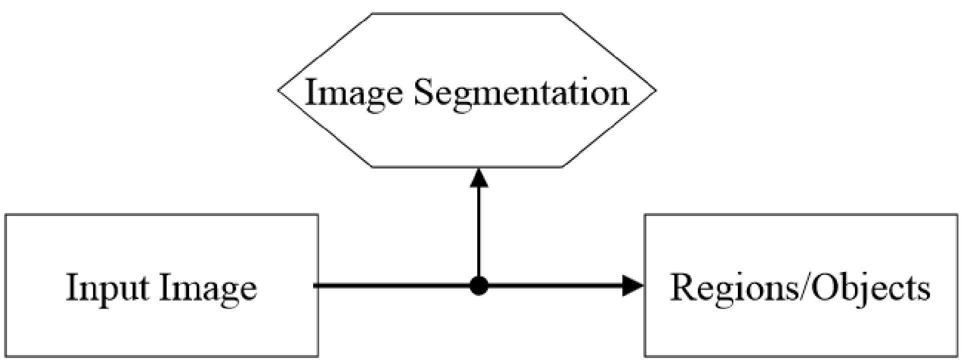
- Analog to digital conversion
- Remove noise/patterns, improve contrast
- Find regions (objects) in the image
- Take measurements of objects/relationships
- Match the above description with similar description of known objects (models)

Image Processing



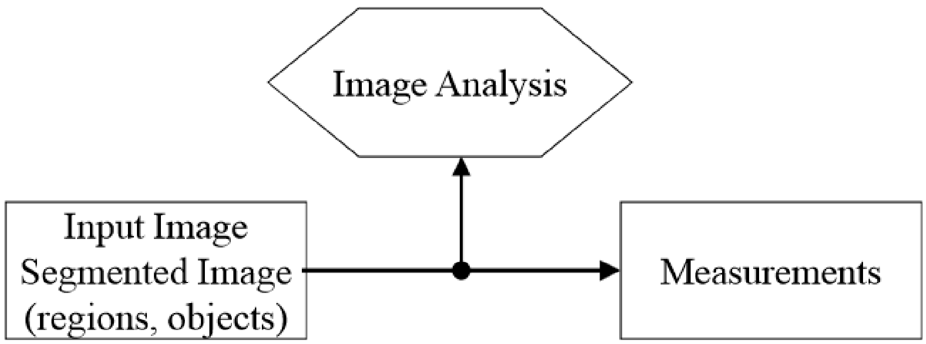
- Image transformation
  - image enhancement (filtering, edge detection, surface detection, computation of depth).
  - Image restoration (remove point/pattern degradation: there exist a mathematical expression of the type of degradation like e.g. Added multiplicative noise, sin/cos pattern degradation etc).

Image Segmentation



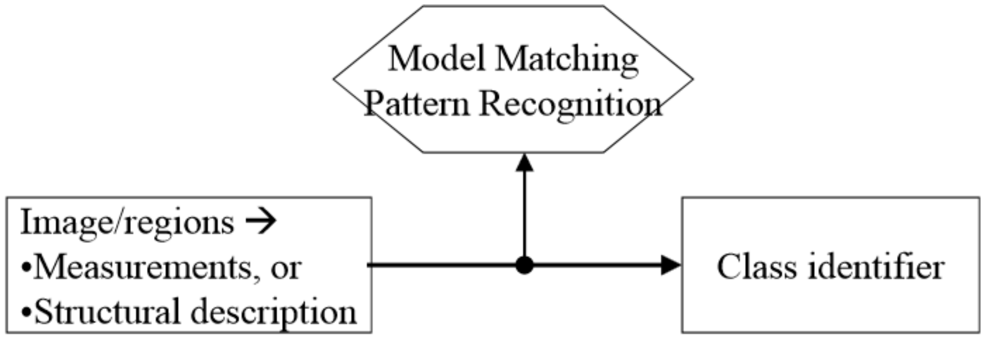
- Classify pixels into groups (regions/objects of interest) sharing common characteristics.
  - Intensity/Color, texture, motion etc.
- Two types of techniques:
  - **Region segmentation**: find the pixels of a region.
  - **Edge segmentation**: find the pixels of its outline contour.

Image Analysis



- Take useful measurements from pixels, regions, spatial relationships, motion etc.
  - Grey scale / color intensity values;
  - Size, distance;
  - Velocity;

Pattern Recognition



- Classify an image (region) into one of a number of known classes
  - Statistical pattern recognition (the measurements form vectors which are classified into classes);
  - Structural pattern recognition (decompose the image into primitive structures).

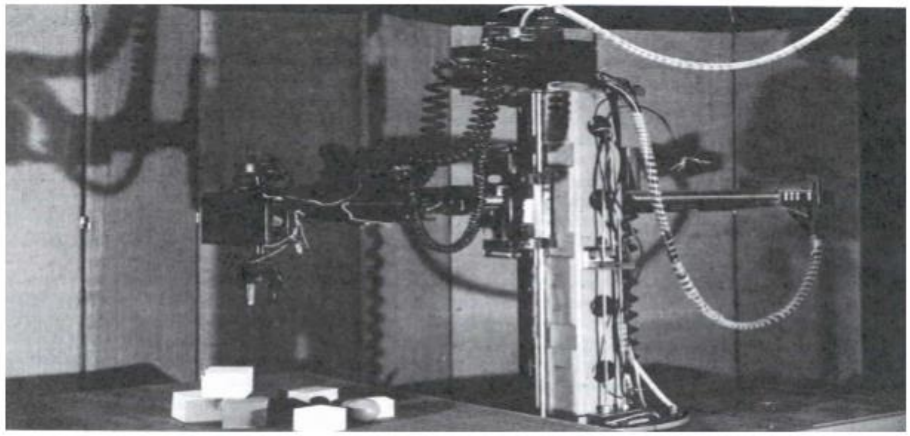
Relationships to other fields

- Image Processing (IP)
- Pattern Recognition (PR)
- Computer Graphics (CG)
- Artificial Intelligence (AI)
- Neural Networks (NN)

Machine Vision Applications

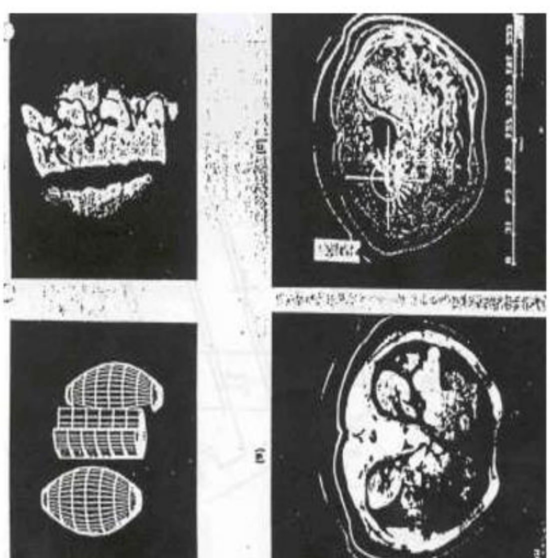
- Robotics
- Medicine
- Remote Sensing
- Meteorology
- Quality inspection

Robot Vision



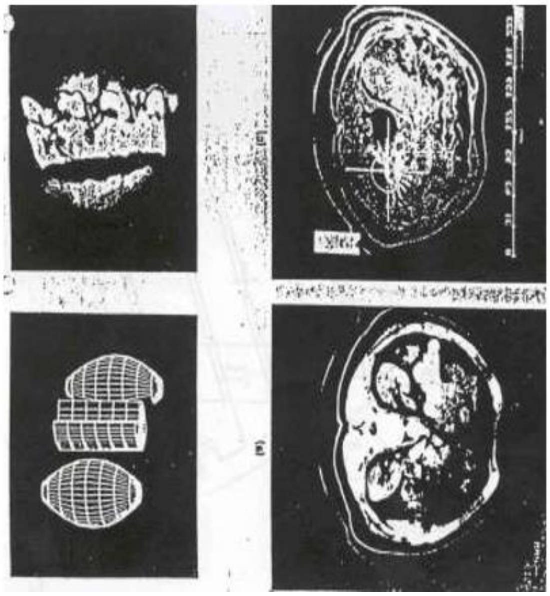
- Machine vision can make a robot manipulator much more versatile.
  - Allow it to deal with variations in parts position and orientation.

Medical Applications



- Assist a physician to reach a diagnosis.
- Construct 2D, 3D anatomy models of the human body.
  - CG geometric models.
- Analyze the image to extract useful features.

Medical Applications



- Assist a physician to reach a diagnosis.
- Construct 2D, 3D anatomy models of the human body.
  - CG geometric models.
- Analyze the image to extract useful features.

COMPONENTS OF MACHINE VISION:

The major components of a machine vision system include the *lighting, lens, image sensor, vision processing, and communications*.

Lighting illuminates the part to be inspected allowing its features to stand out so they can be clearly seen by camera.

The various components of a machine vision system include:

Lighting
Lenses
Vision Processing
Image Sensor
Communications

What Is Robotics?

Robotics is the intersection of science, engineering and technology that produces machines, called robots, that substitute for (or replicate) human actions.

Main components of a robot:

Robots are built to present solutions to a variety of needs and fulfill several different purposes, and therefore, require a variety of specialized components to complete these tasks. However, there are several components that are central to every robot's construction, like a power source or a central processing unit.

MAIN COMPONENTS OF A ROBOT are:

Control system
Sensors
Actuators
Power Supply
End Effectors

Control system

Computation includes all of the components that make up a robot's central processing unit, often referred to as its control system. Control systems are programmed to tell a robot how to utilize its specific components, similar in some ways to how the human brain sends signals throughout the body, in order to complete a specific task.

Sensors

Sensors provide a robot with stimuli in the form of electrical signals that are processed by the controller and allow the robot to interact with the outside world. Common sensors found within robots include video cameras that function as eyes, photo resistors that react to light and microphones that operate like ears.

Actuators

Actuators are the components that are responsible for this movement. These components are made up of motors that receive signals from the control system and move in tandem to carry out the movement necessary to complete the assigned task. Actuators can be made of a variety of materials, such as metal or elastic, and are commonly operated by use of compressed air (pneumatic actuators) or oil (hydraulic actuators,) but come in a variety of formats to best fulfill their specialized roles.

Power Supply

Like the human body requires food in order to function, robots require power. Stationary robots, such as those found in a factory, may run on AC power through a wall outlet but more commonly, robots operate via an internal battery.

End Effectors

End effectors are the physical, typically external components that allow robots to finish carrying out their tasks. Robots in factories often have interchangeable tools like paint sprayers and drills, surgical robots may be equipped with scalpels and other kinds of robots can be built with gripping claws or even hands for tasks like deliveries, packing, bomb diffusion and much more.

APPLICATIONS OF ROBOTICS

- Helping fight forest fires
- Working alongside humans in manufacturing plants (known as co-bots)

- Robots that offer companionship to elderly individuals
- Surgical assistants
- Last-mile package and food order delivery
- Autonomous household robots that carry out tasks like vacuuming and mowing the grass
- Assisting with finding items and carrying them throughout warehouses
- Used during search-and-rescue missions after natural disasters
- Landmine detectors in war zones

Robotics:Perception

In robotics, perception is understood as a system that endows the robot with the ability to perceive, comprehend, and reason about the surrounding environment. The key components of a perception system are essentially sensory data processing, data representation (environment modeling), and ML-based algorithms, as illustrated in Figure 1. Since *strong AI* is still far from being achieved in real-world robotics applications.

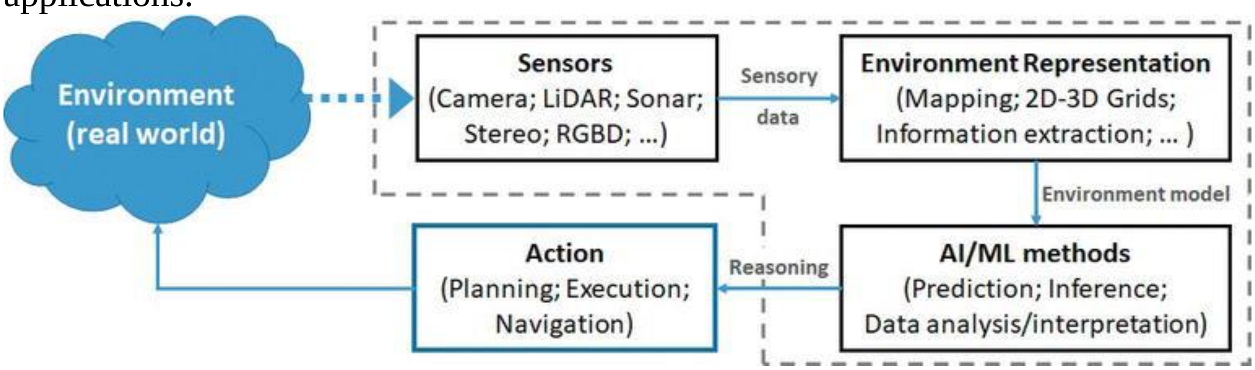


Figure 1.

Key modules of a typical robotic perception system: sensory data processing (focusing here on visual and range perception); data representations specific for the tasks at hand; algorithms for data analysis and interpretation (using AI/ML methods); and planning and execution of actions for robot-environment interaction.

Robotic perception is crucial for a robot to make decisions, plan, and operate in real-world environments, by means of numerous functionalities and operations from occupancy grid mapping to object detection.

Some examples *of robotic perception subareas, including autonomous robot-vehicles, are obstacle detection*, object recognition, semantic place classification, voice recognition, activity classification, road detection, vehicle detection, pedestrian detection, object tracking, human detection, and environment change detection.

Nowadays, most of robotic perception systems use machine learning (ML) techniques, ranging from classical to deep-learning approaches. Machine learning for robotic perception can be in the form of unsupervised learning, or supervised classifiers using handcrafted features, or deep-learning neural networks (e.g., convolutional neural network (CNN)), or even a combination of multiple methods.