REVA UNIVERSITY
Bengaluru, India

Artificial Intelligence

M21DES221 – Academic Year 2024-25 – Sem II Even Semester

MCA , School of CSA.

www.reva.edu.in

Shreetha Bhat
Assistant Professor

Agenda

- Quick look on Syllabus

Quick Look on Syllabus

Unit 4 – 13 Hrs

Fuzzy Logic Systems, Expert Systems, Natural Language Processing

Fuzzy Logic Systems: Introduction; Crisp Sets; Fuzzy Sets; Fuzzy Terminology; Fuzzy Logic Control-Fuzzy Room Cooler.

Expert Systems: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.

Natural Language Processing: Definition, History of NLP, Advantages and Disadvantages of NLP, Components of NLP, real time Applications, NLP pipeline, Phases of NLP, Difficulties in NLP, NLP APIs

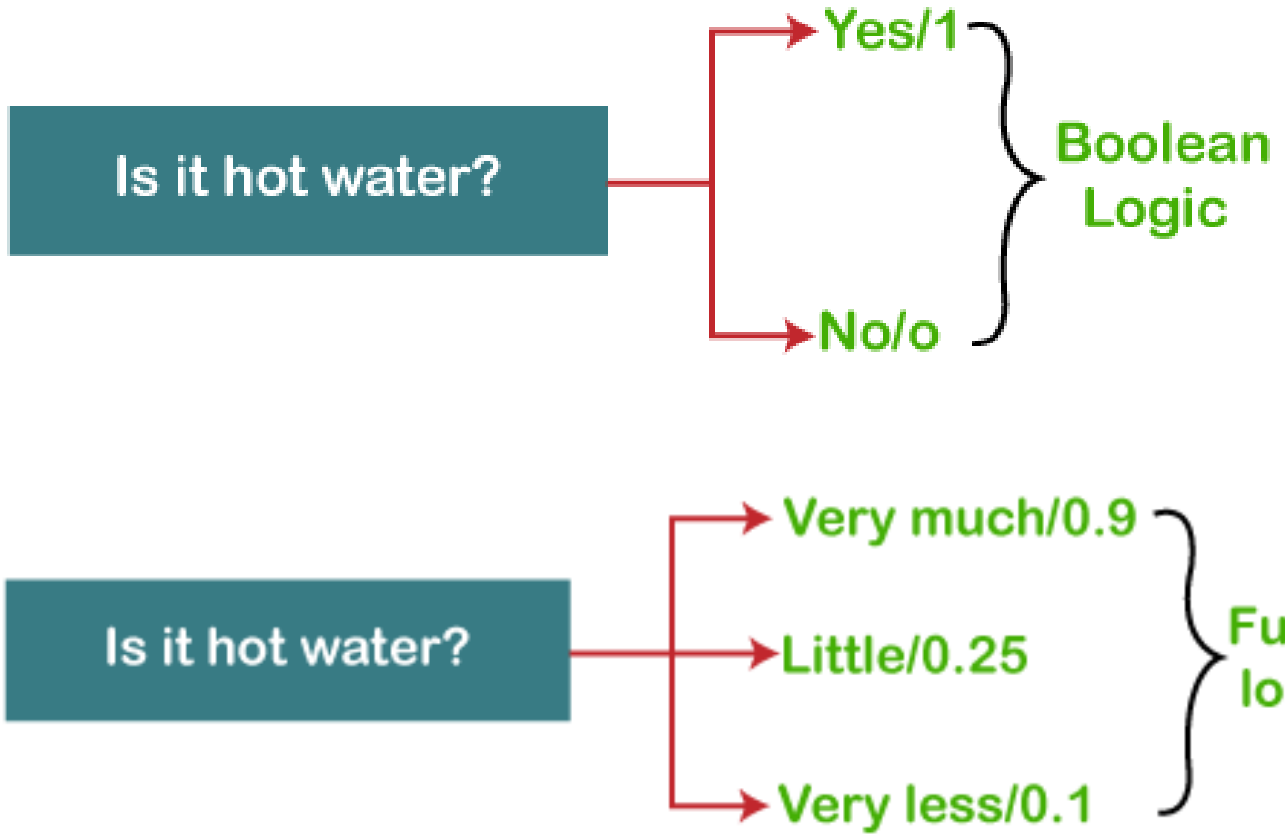
Fuzzy Logic Systems - Introduction

The term fuzzy refers to things that are not clear or are vague.

- Fuzzy logic provides very valuable flexibility for reasoning.
- Using Fuzzy Logic, we can consider the inaccuracies and uncertainties of any situation.
- In the fuzzy system, there is no logic for the absolute truth and absolute false value.
- There is an intermediate value too present which is partially true and partially false.

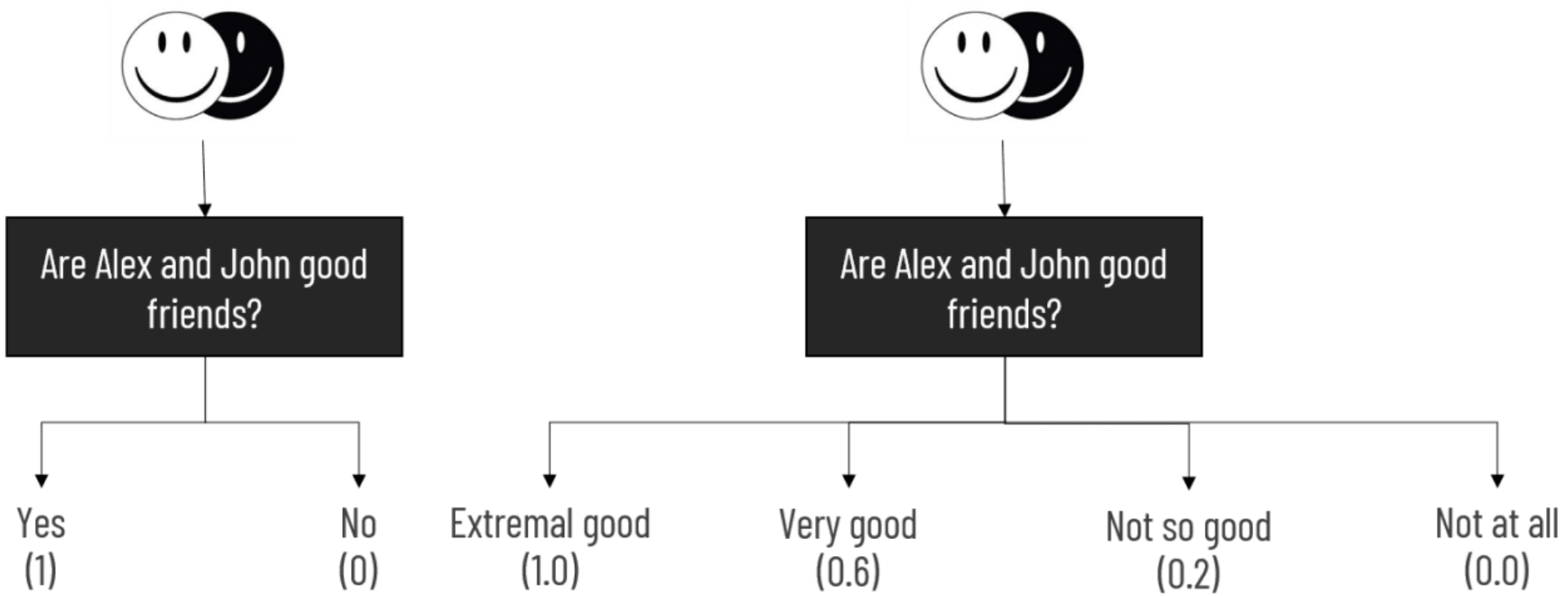
Fuzzy Logic Systems - Introduction

Example – Boolean V/s Fuzzy Logic



Fuzzy Logic Systems - Introduction

Example – Boolean V/s Fuzzy Logic



Fuzzy Logic Systems - Advantages

- This system can work with any type of inputs whether it is imprecise, distorted or noisy input information.
- The construction of Fuzzy Logic Systems is easy and understandable.
- Fuzzy logic comes with mathematical concepts of set theory and the reasoning of that is quite simple.
- It provides a very efficient solution to complex problems in all fields of life as it resembles human reasoning and decision-making.
- The algorithms can be described with little data, so little memory is required.

Fuzzy Logic Systems - Disadvantages

- Many researchers proposed different ways to solve a given problem through fuzzy logic which leads to ambiguity. There is no systematic approach to solve a given problem through fuzzy logic.
- Proof of its characteristics is difficult or impossible in most cases because every time we do not get a mathematical description of our approach.
- As fuzzy logic works on precise as well as imprecise data so most of the time accuracy is compromised.

Fuzzy Logic Systems - Crisp sets

- Crisp set is a collection of **unordered distinct** elements, which are derived from Universal set.
- **Universal set consists of all possible elements which take part in any experiment.**
- **Set is quite useful and important way of representing data.**

Fuzzy Logic Systems - Crisp sets

Let X represents a set of natural numbers, so

$X = \{1, 2, 3, 4, \ldots\}$

Sets are always defined with respect to some universal set. Let us derive two sets A and B from this universal set X.

$A = Set\ of\ even\ numbers = \{2, 4, 6, \ldots\}$

$B = Set\ of\ odd\ number = \{1, 3, 5, \ldots\}$

Fuzzy Logic Systems - Crisp sets

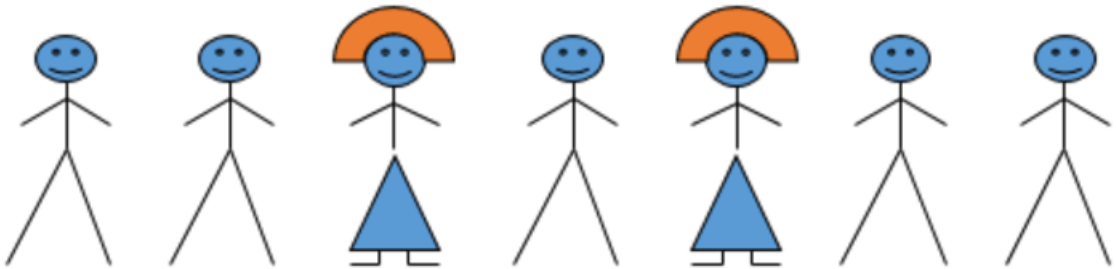
Example of Crisp Set

Consider X represents a class of students that acts as the Universe of discourse. If you ask, "Who does have a driving license?" All students might not have a driving license. So those students who have driving licenses will have a *membership value of 1* for this particular set and the rest of them will have a *membership value of zero.*

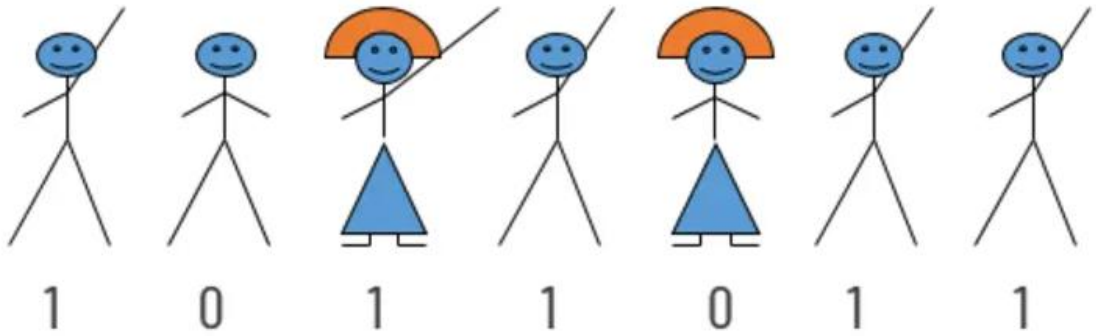
We can define set A as equal to the set of students having driving licenses and A will be a subset of universal set X.

Fuzzy Logic Systems - Crisp sets

Example of Crisp Set



Universal set



Crisp Set

Fuzzy Logic Systems – Fuzzy Sets

Fuzzy sets introduce a certain amount of vagueness to reduce complexity of comprehension. This set consists of elements that signify the degree or grade of membership to a fuzzy aspect. Membership values usually use closed intervals and denote the sense of belonging of a member of a crisp set to a fuzzy set. To make the point clear consider a crisp set A comprising of elements that signify the ages of a set of people in years.

$$A = \{2,4,10,15,21,30,35,40,45,60,70\}$$

We could classify age in terms of what are known as fuzzy linguistic variables – *infant*, *child*, *adolescent*, *adult*, *young* and *old*. A person whose age is 15 is no doubt young but how would you categorize a person who is 30. If the latter is to be considered *young* what about the person who is 40? Is he old? How do we translate all these into numbers for efficiently making the computer understand what our feelings about age are?

Fuzzy Logic Systems – Fuzzy Sets

Table 22.1 *Ages and their memberships*

Age	*Infant*	*Child*	*Adolescent*	*Young*	*Adult*	*Old*
2	1	0	0	1	0	0
4	0.1	0.5	0	1	0	0
10	0	1	0.3	1	0	0
15	0	0.8	1	1	0	0
21	0	0	0.1	1	0.8	0.1
30	0	0	0	0.6	1	0.3
35	0	0	0	0.5	1	0.35
40	0	0	0	0.4	1	0.4
45	0	0	0	0.2	1	0.6
60	0	0	0	0	1	0.8
70	0	0	0	0	1	1

The values in the table indicate memberships to the fuzzy sets – *infant, child, adolescent, young, adult* and *old*. Thus a child of age 4 belongs only 50% to the fuzzy set *child* while when he is 10 years he is a 100% member. Note that membership is different from probabilities. Memberships do not necessarily add up to 1. The entries in the table have been made after a manual evaluation of the different ages.

Fuzzy Terminology

Universe of Discourse (U):

This is defined as the range of all possible values that comprise the input to the fuzzy system.

Fuzzy Set

Any set that empowers its members to have different grades of membership (based on a membership function) in an interval [0,1] is a fuzzy set.

Membership function

The membership function μ_A which forms the basis of a fuzzy set is given by

$$\mu_A: U \rightarrow [0,1]$$

where the closed interval is one that holds real numbers.

Fuzzy Terminology

Support of a fuzzy set (S_f)

The support S of a fuzzy set f, in a universal crisp set U is that set which contains all elements of the set U that have a non-zero membership value in f. For instance, the support of the fuzzy set *adult* is

$$S_{adult} = \{21,30,35,40,45,60,70\}$$

Depiction of a fuzzy set

A fuzzy set f in a universal crisp set U, is written as

$$f = \mu_1 / s_1 + \mu_2 / s_2 + \mu_3 / s_3 + \ldots + \mu_n / s_n$$

where μ_i is the membership and s_i is the corresponding term in the *support* set of f i.e. S_f.

This is however only a representation and has *no algebraic implication* (the slash and + signs do not have any meaning).

Accordingly,

$$\text{Old} = 0.1/21 + 0.3/30 + 0.35/35 + 0.4/40 + 0.6/45 + 0.8/60 + 1/70$$

Fuzzy Terminology

Fuzzy Set Operations

- **Union:** The membership function of the union of two fuzzy sets A and B is defined as the maximum of the two individual membership functions. It is equivalent to the Boolean OR operation.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

- **Intersection:** The membership function of the intersection of two fuzzy sets A and B is defined as the minimum of the two individual membership functions and is equivalent to the Boolean AND operation.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

- **Complement:** The membership function of the complement of a fuzzy set A is defined as the negation of the specified membership function: $\mu_{\bar{A}}$. This is equivalent to the Boolean NOT operation

$$\mu_{\bar{A}} = \mu_{A \cup B} = (1 - \mu_A)$$

It may be further noted here that the laws of Associativity, Commutativity , Distributivity and De Morgan's laws hold in fuzzy set theory too.

Fuzzy Logic Control

Fuzzy logic is a technique used in a variety of applications, from household appliances to complex systems.

It is not inherently intelligent but can be combined with learning and adaptation techniques to create intelligent systems.

Fuzzy Logic Control

We discuss how fuzzy logic can be used to control the speed of a motor based on temperature and humidity.

This technique is particularly relevant for room coolers used in tropical environments.

Fuzzy Logic Control

We assume a traditional room cooler setup using a fan encased in a moist material, controlled by sensors and a motorized pump to regulate water flow and fan speed.

Key Points:

- Traditional room cooler design with fan and moist material.
- Motorized pump controls water flow.
- Sensors monitor fan speed and room temperature.
- Fan speed can be manual or automatic.
- Focus on smooth control and water conservation.

Fuzzy Logic Control

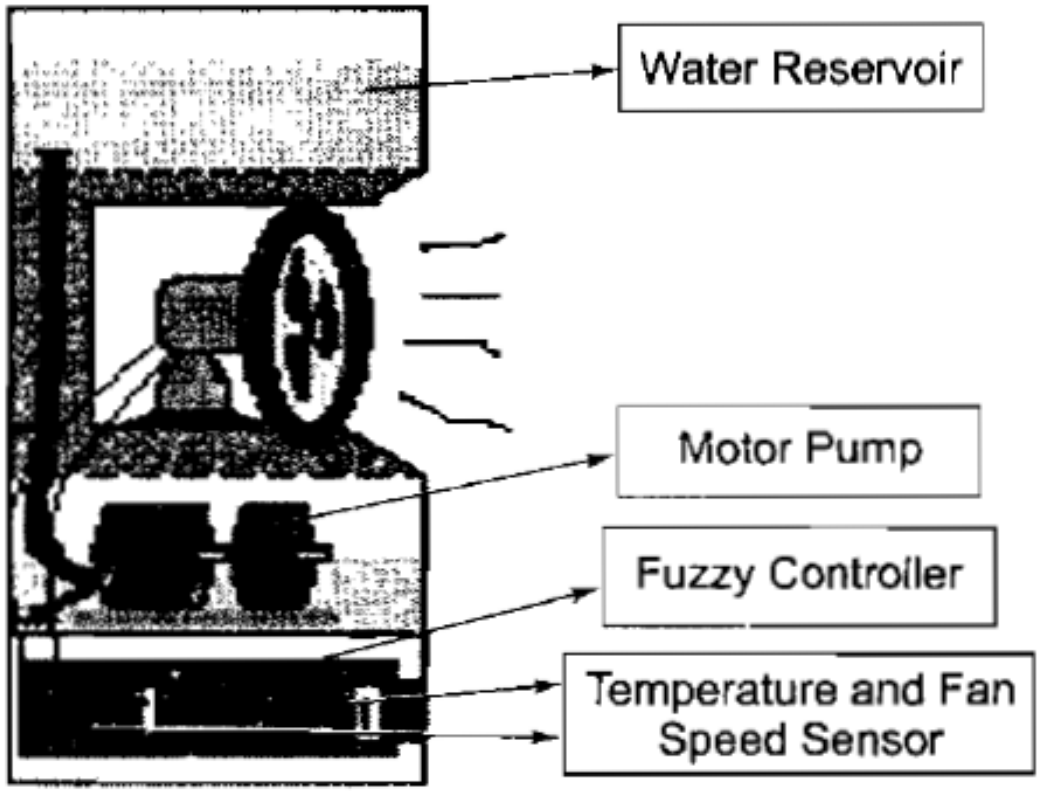


Fig. 22.1 *A Sectional View of a Fuzzy Room Cooler*

Fuzzy Logic Control

For simplicity to maintain room temperature, only the water flow rate needs to be adjusted based on fan speed and temperature.

The next step is to design a fuzzy engine to control this system.

Fuzzy Logic Control

Fuzzy Regions

- Two parameters (temperature and pressure) determine the water flow rate.
- Fuzzy terms are defined for temperature: Cold, Cool, Moderate, Warm, and Hot.
- Fuzzy terms are defined for fan speed: Slack, Low, Medium, Brisk, and Fast.
- Temperature and fan speed can be expressed using these fuzzy terms.
- The water flow rate, controlled by the motorized pump, is also defined using fuzzy terms: Strong-Negative (SN), Negative (N), Low-Negative (LN), Medium (M), Low-Positive (LP), Positive (P), and High-Positive (HP).

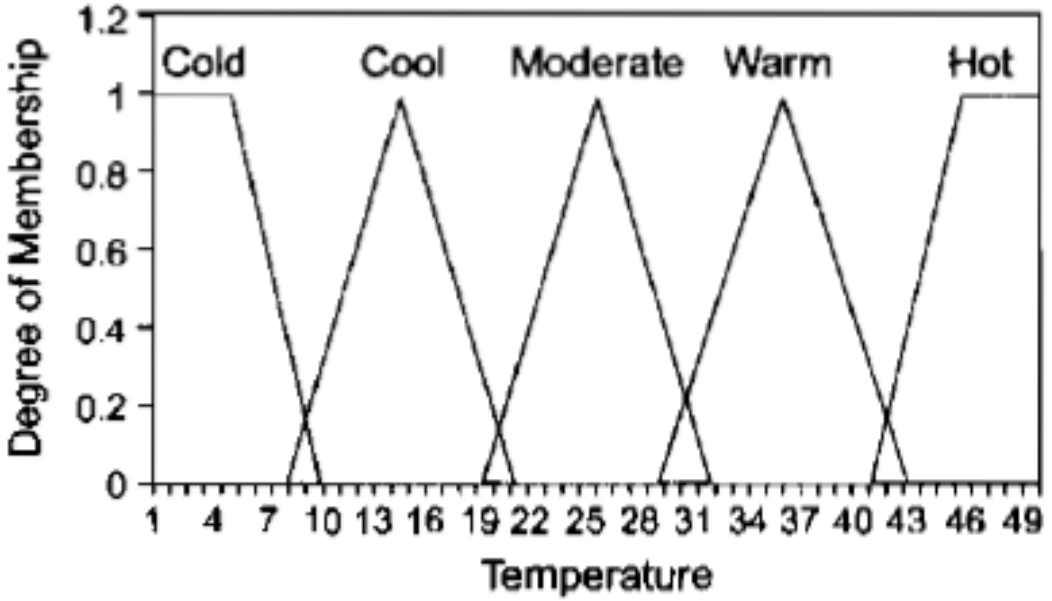
Fuzzy Logic Control

Fuzzy Profile

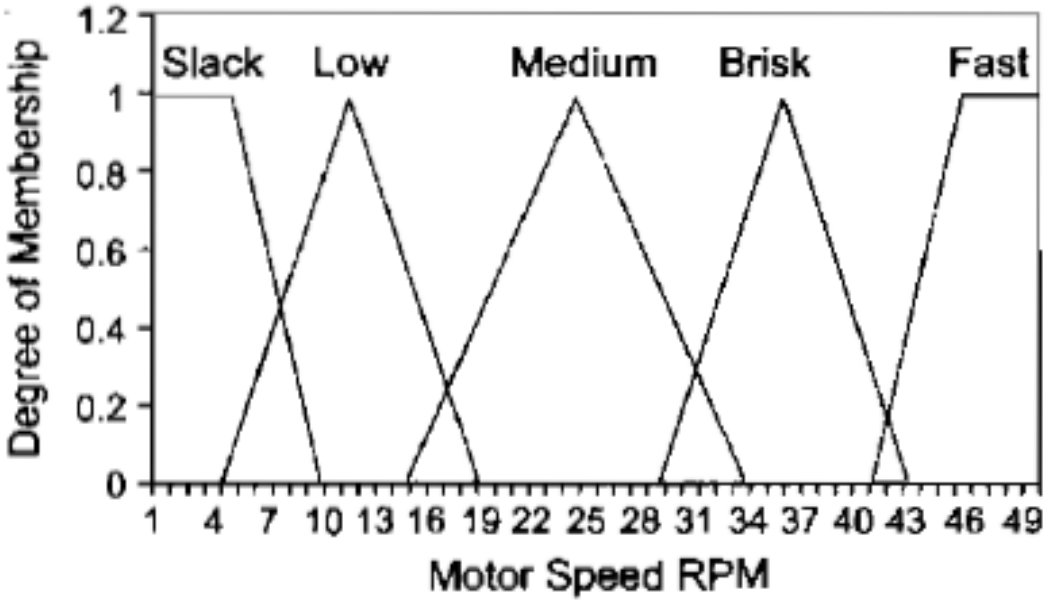
- Real data is used to define profiles for temperature, fan motor speed, and flow rate.
- Membership functions are assigned to each parameter's values.
- The regions defined for each fuzzy set overlap.
- Membership values change gradually between sets.
- For example, at 25 degrees, membership in "moderate" is high, but at 30 degrees, it decreases, and membership in "warm" increases.
- Profiles must be carefully designed based on the system's nature and desired behavior.
- Graphs in Fig. 22.2 and Fig. 22.3 depict these membership functions.

Fuzzy Logic Control

Fuzzy Profile



(a) Temperature



(b) Fan Motor Speed

Fig. 22.2

Fuzzy Logic Control

Fuzzy Profile

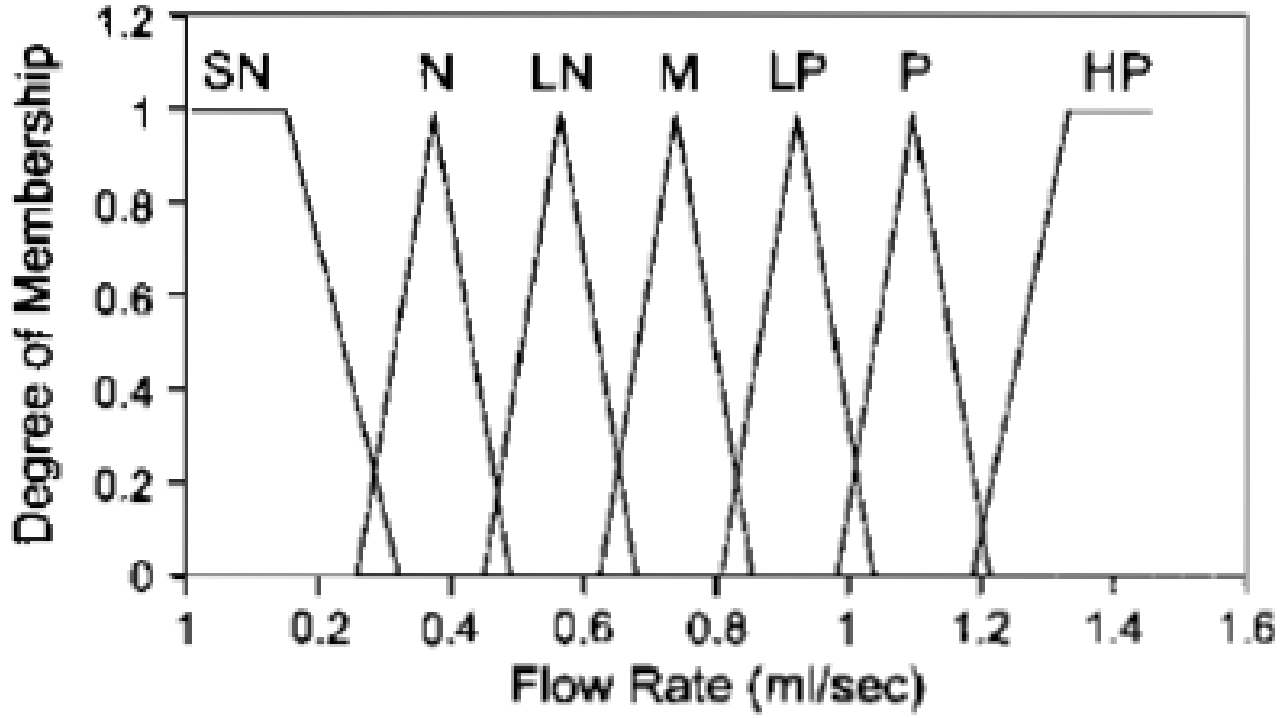


Fig. 22.3 *Water Flow Rate*

Fuzzy Logic Control

Fuzzy Rules

R1: If temperature is HOT ***and*** fan motor speed is SLACK then flow-rate is HIGH-POSITIVE.
R2: If temperature is HOT ***and*** fan motor speed is LOW then flow-rate is HIGH-POSITIVE
R3: If temperature is HOT ***and*** fan motor speed is MEDIUM then the flow-rate is POSITIVE.
R4: If temperature is HOT ***and*** fan motor speed is BRISK then the flow-rate is HIGH-POSITIVE.
R5: If temperature is WARM ***and*** fan motor speed is MEDIUM then the flow-rate is LOW-POSITIVE,
R6: If temperature is WARM ***and*** fan motor speed is BRISK then the flow-rate is POSITIVE.
R7: If temperature is COOL ***and*** fan motor speed is LOW then flow-rate is NEGATIVE.
R8: If temperature is MODERATE ***and*** fan motor speed is LOW then flow-rate is MEDIUM.

Fuzzy Logic Control

Fuzzification

- The fuzzifier is the core of the fuzzy engine.
- Sensor values for temperature and fan speed are mapped to fuzzy regions based on their membership functions.
- For example, if the temperature is 42 degrees and fan speed is 31 rpm (**revolutions per minute**), their corresponding membership values and fuzzy regions are determined.

Fuzzy Logic Control

Fuzzification

Parameter	*Fuzzy Regions*	*Memberships*
Temperature	warm, hot	0.142, 0.2
Fan speed	medium, brisk	0.25,0.286

- Both temperature and fan speed belong to two fuzzy regions each.
- Rules R3, R4, R5, and R6 are applicable in this situation.
- There is a conflict between the rules:
- Two rules suggest a POSITIVE flow rate.
- Two other rules suggest a LOW-POSITIVE or HIGH-POSITIVE flow rate.
- While the possible flow rates have been identified, the exact crisp value is still unknown.

Fuzzy Logic Control

Defuzzifier

- Defuzzification is the process of converting fuzzy outputs (like LOW-POSITIVE, POSITIVE, and HIGH-POSITIVE) into a single crisp value.
- Defuzzification is necessary to control the pump's actuator.
- Common defuzzification methods include Centre of Gravity and Composite Maxima.
- Both methods involve calculating a composite region formed by portions A, B, C, and D on the output profile.
- For parameters connected by AND, the minimum membership value is used.

Fuzzy Logic Control

Defuzzifier

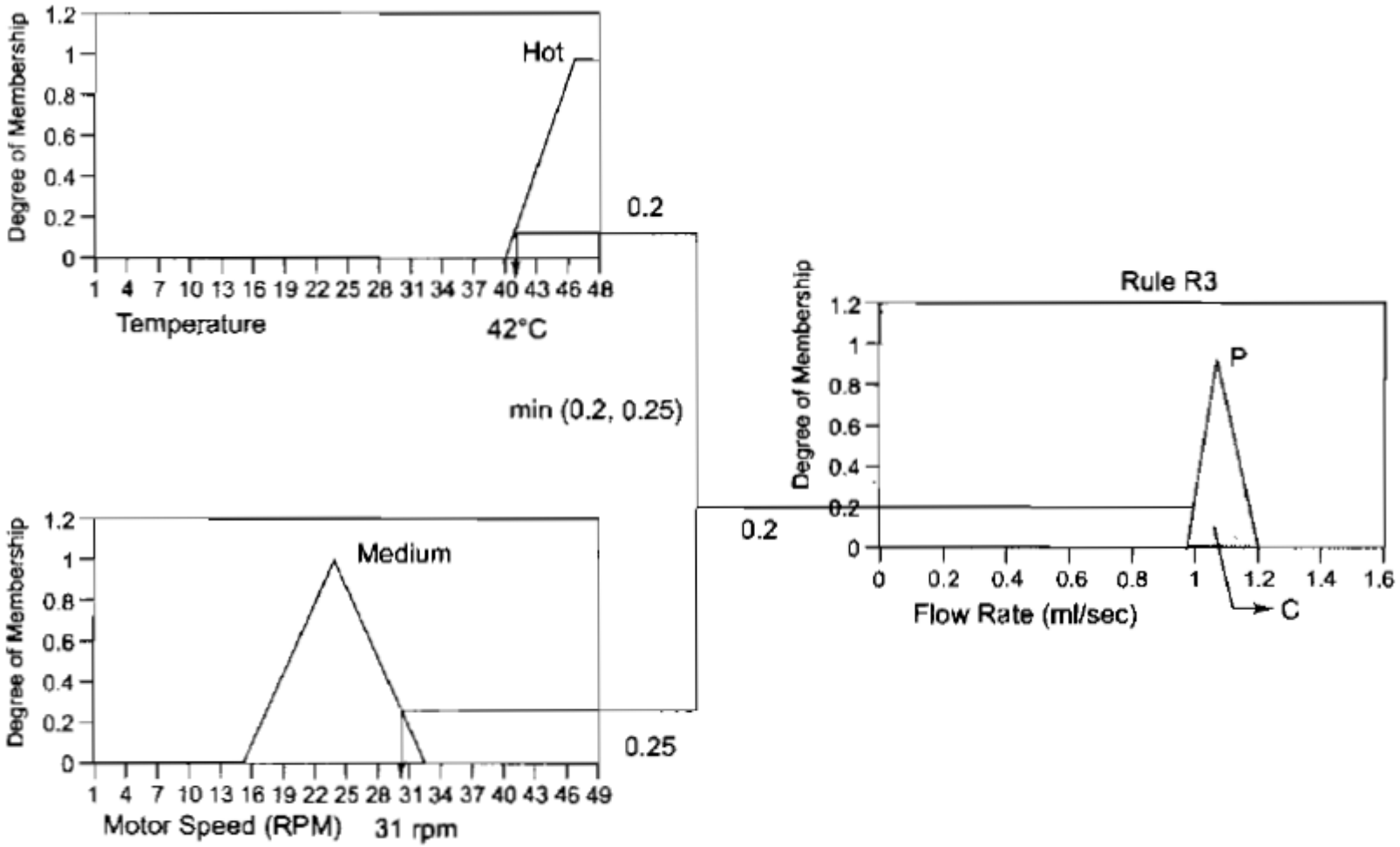


Fig. 22.4 *Defuzzification (contd.)*

Fuzzy Logic Control

Defuzzifier

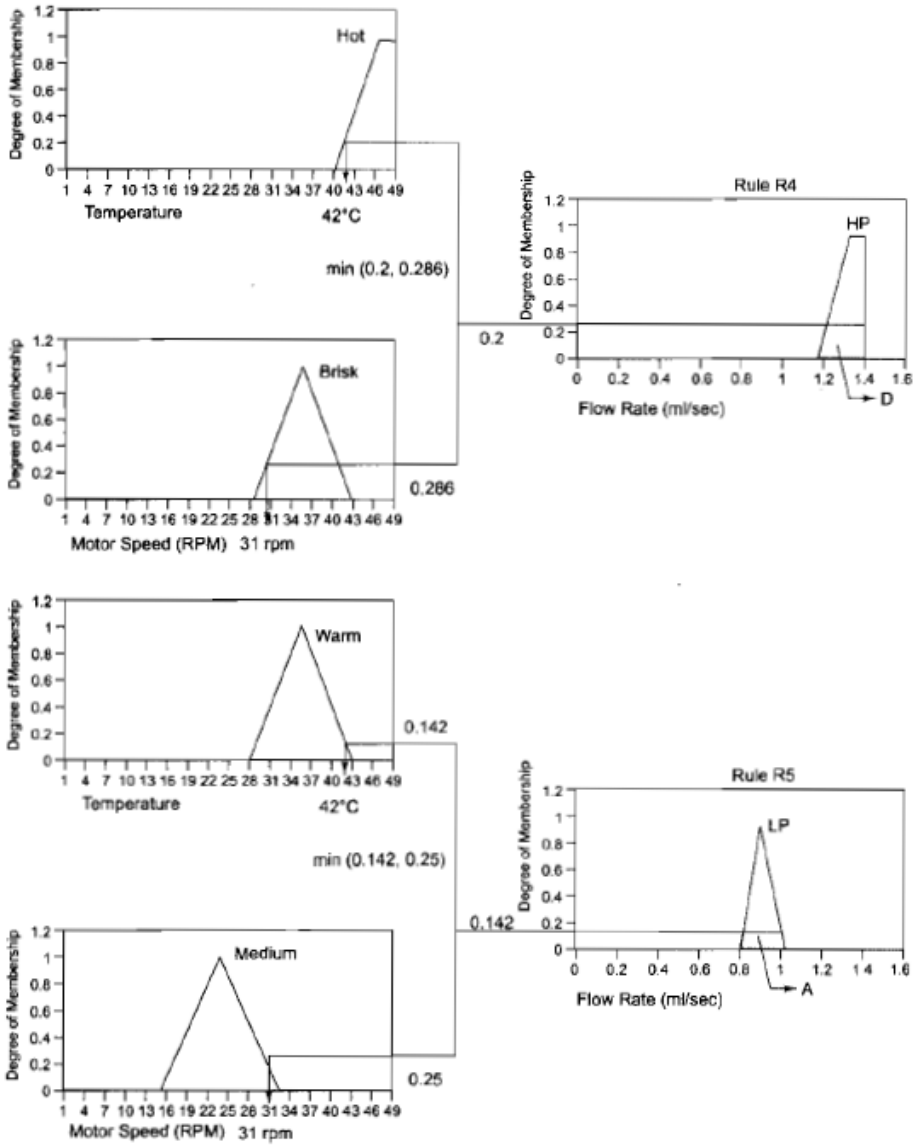
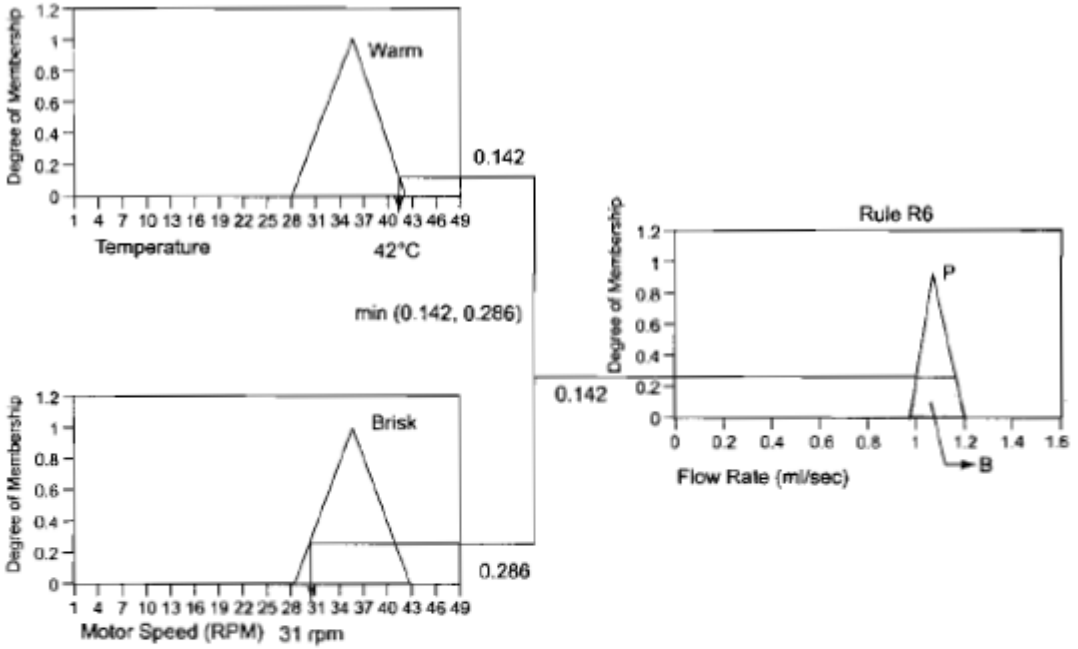


Fig. 22.4 *Defuzzification (contd.)*

Fuzzy Logic Control

Defuzzifier



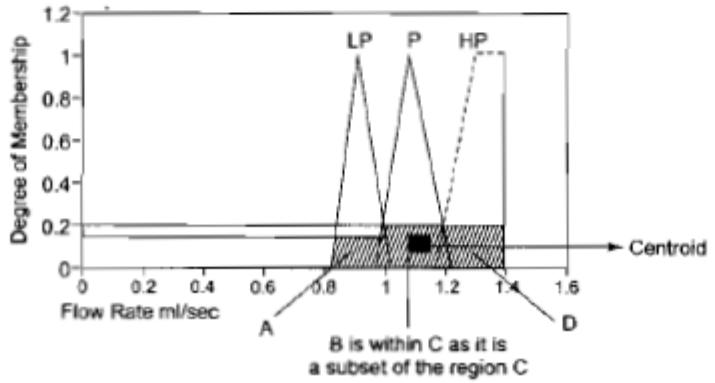


Fig. 22.4 *Defuzzification*

Fuzzy Logic Control

Defuzzifier

- A horizontal line is drawn through the output fuzzy set profile to create a region.
- The maximum membership value is used for "OR" operations, while the minimum is used for "AND" operations.
- All output surfaces are combined to form the composite output region.
- The Centre of Gravity or Composite Maxima of this region is used as the crisp output.
- Centre of Gravity is usually preferred for control applications.
- The crisp output (X-coordinate of the Centroid) determines the desired flow rate for the pump.

Expert Systems - Introduction

- Expert systems solve problems that are normally solved by human experts.
- They require a substantial domain knowledge base and reasoning mechanisms.
- They need a mechanism to explain their solutions to users.
- Expert systems represent applied AI in a broad sense.
- They are currently lagging behind research advances but will eventually catch up.

Expert Systems - Introduction

- Expert systems deal with diverse problems across various domains.
- There are general issues and specific techniques for different problem classes.
- Problem classification plays an important role in designing problem-solving systems.
- Tools developed for one task can often be useful for similar tasks.

Expert Systems -

Representing and Using Domain Knowledge

- Expert systems are complex AI programs that leverage various techniques.
- Production rules are a common way to represent domain knowledge.
- These rules are often coupled with frame systems that define objects.
- The MYCIN system is an example of an expert system using rules.
- Different expert systems use rules in different ways, illustrating variations in their operation.

Expert Systems -

Representing and Using Domain Knowledge

R1 [McDermott, 1982; McDermott, 19841 (sometimes also called XCON) is a program that configures DEC VAX systems. Its rules look like this:

```
If: the most current active context is distributing
        massbus devices, and
    there is a single-port disk drive that has not been
        assigned to a massbus, and
    there are no unassigned dual-port disk drives, and
        the number of devices that each massbus should
        support is known, and
    there is a massbus that has been assigned at least
        one disk drive and that should support additional
        disk drives,
    and the type of cable needed to connect the disk drive
        to the previous device on the massbus is known
then: assign the disk drive to the massbus.
```

Expert Systems -

Representing and Using Domain Knowledge

- R1's rules do not contain numeric measures of certainty, unlike MYCIN's.
- In R1's task domain, it is possible to precisely state the correct action for each set of circumstances, even if it requires complex antecedents.
- This is due to the abundance of human expertise in this area.
- Since R1 is a design task, it is not necessary to consider all alternatives; one good solution is enough.
- As a result, probabilistic information is not required in R1.

Expert Systems -

Expert System Shells

- **Initial development:** Early expert systems were built from scratch, often in LISP.
- **Commonalities:** After building several systems, it became clear that they shared common characteristics, particularly the use of declarative representations and an interpreter.
- **Shell concept:** This led to the idea of separating the interpreter from the domain-specific knowledge, creating shells that could be used to build new expert systems.
- **EMYCIN:** An influential example of a shell derived from MYCIN is EMYCIN.

Expert Systems -

Expert System Shells

- **Commercial shells:** Several commercially available shells exist today, offering more flexibility and support for various reasoning mechanisms than MYCIN.
- **Evolution of shells:** Early shells focused on knowledge representation, reasoning, and explanation. Later, knowledge acquisition tools were added.
- **Integration needs:** As expert systems became more complex, the need to integrate them with other systems and programs became evident.
- **Shell requirements:** Shells must provide an easy-to-use interface to facilitate integration with other programming environments.

Expert Systems -

Explanation

- **Effective interaction:** Expert systems need to be able to interact easily with people.
- **Explainability:** They must be able to explain their reasoning process, especially in domains like medicine where accountability is crucial.
- **Learning:** Expert systems should be able to acquire new knowledge and modify existing knowledge, either through interaction with experts or by learning from data.

Expert Systems -

Knowledge Acquisition

- **Building expert systems:** Knowledge engineers interview domain experts to extract knowledge, which is then translated into rules.
- **Iterative refinement:** The system is refined until it reaches expert-level performance.
- **Automation:** The process is time-consuming, so there's a focus on automating knowledge acquisition.
- **Knowledge acquisition programs:** These programs support entering, maintaining, and ensuring consistency of knowledge bases.

Expert Systems -

Knowledge Acquisition

- **Problem-solving paradigms:** Programs tailored to specific paradigms (e.g., diagnosis or design) are more effective.
- **MOLE:** A knowledge acquisition system for heuristic classification problems, used with the cover-and-differentiate method.
- **MOLE's process:** It takes input data, generates candidate explanations, and uses differentiating knowledge to select the best one.
- **Iterative process:** Explanations are justified until ultimate causes are identified.

Expert Systems -

Knowledge Acquisition

- **MOLE:** A knowledge acquisition system that interacts with a domain expert to build a knowledge base.
- **Knowledge Base Construction:**
- **Initial Phase:** MOLE asks the expert to list common symptoms and their possible explanations.
- **Iterative Refinement:** MOLE seeks higher-level explanations and builds an influence network to represent relationships between symptoms and causes.
- **Covering and Anticipatory Knowledge:** MOLE identifies conditions under which a hypothesis is correct and determines if a symptom will appear if the hypothesis is true.

Expert Systems -

Knowledge Acquisition

Refinement of the Knowledge Base:

- **Identifying Weaknesses:** MOLE looks for holes in the knowledge base and prompts the expert to fill them.
- **Learning from Mistakes:** MOLE-p (the performance system) solves problems, and the expert corrects any errors.
- **Improving Explanations:** MOLE identifies reasons for incorrect diagnoses, such as missing or incorrect knowledge.

Expert Systems -

Knowledge Acquisition

Limitations and Applications:

- **Pre-enumeration:** MOLE requires that possible solutions or classifications can be pre-enumerated.
- **Knowledge Encoding:** Knowledge must be encodable in terms of covering and differentiating relationships.
- **Applications:** MOLE has been used in domains like car engine diagnosis, steel-rolling mill problems, and power plant efficiency.

Natural Language Processing

- **Natural Language Processing (NLP)** is a field within Artificial Intelligence (AI) that focuses on the interaction between computers and human (natural) languages.
- The goal of NLP is to enable machines to understand, interpret, and respond to human language in a meaningful way. This involves tasks such as reading text, listening to speech, and even generating responses.

History of NLP

1. **1940s–1950s: Early Foundations**
    - Alan Turing proposed the Turing Test (1950).
    - Rule-based systems like the **Georgetown Experiment** translated languages using hand-coded rules.
2. **1950s–1960s: Linguistic Models**
    - Introduction of syntax-based approaches like **Chomsky's Context-Free Grammar**.
    - Early parsing algorithms (e.g., CYK) were developed.
3. **1970s–1980s: Statistical Methods**
    - Shift from rule-based to **statistical models** (e.g., Hidden Markov Models for speech recognition).
    - Rise of machine translation systems (e.g., IBM's statistical translation model).

History of NLP

4. **1990s–2000s: Machine Learning**
    - Adoption of machine learning techniques like **Support Vector Machines** and **Naive Bayes**.
    - Development of tools like **WordNet** and NLP libraries.
5. **2010s–Present: Deep Learning and Transformers**
    - Revolutionized by **transformer models** like BERT and GPT.
    - Applications expanded to virtual assistants, sentiment analysis, and more advanced tasks.

NLP - Advantages

- **Efficient Text Processing**: Automates large-scale text analysis (e.g., customer feedback, social media) to save time and resources.
- **Improved Communication**: Powers chatbots, virtual assistants, and translation tools for seamless human-machine interaction.
- **Enhanced Decision-Making**: Extracts insights from unstructured data, supporting data-driven decisions in businesses.
- **Multilingual Capabilities**: Bridges language barriers through accurate and scalable machine translation.
- **Scalability**: Handles massive datasets in real-time, suitable for industries like healthcare, finance, and e-commerce.

NLP - Disadvantages

- **Contextual Limitations**: Struggles with understanding nuances like sarcasm, idioms, or cultural references.
- **Data Dependency**: Requires vast, diverse datasets to perform accurately, which can be costly and time-consuming to obtain.
- **Bias Issues**: May inherit biases present in the training data, leading to unfair or discriminatory outcomes.
- **Complexity**: Developing and fine-tuning NLP systems often require specialized expertise and computational resources.
- **Privacy Concerns**: Processing sensitive text data (e.g., personal conversations) can raise ethical and privacy challenges

Components of NLP

1. **Text Preprocessing**
    - Prepares raw text for analysis by cleaning and structuring data.
    - Tasks include:
        - **Tokenization**: Splitting text into words or sentences.
        - **Stopword Removal**: Removing common words like "the" or "is".
        - **Stemming and Lemmatization**: Reducing words to their root forms.
2. **Syntactic Analysis (Syntax)**
    - Analyzes the grammatical structure of sentences.
    - Key tasks:
        - **Part-of-Speech (POS) Tagging**: Identifies nouns, verbs, adjectives, etc.
        - **Parsing**: Builds a syntax tree to understand sentence structure.

Components of NLP

3. **Semantic Analysis (Semantics)**
    - Focuses on the meaning of words and sentences.
    - Includes:
        - **Named Entity Recognition (NER)**: Identifies entities like names, places, or dates.
        - **Word Sense Disambiguation**: Resolves ambiguities in word meanings.
4. **Pragmatic Analysis (Context)**
    - Interprets meaning based on context and user intent.
    - Examples: Understanding sarcasm, conversational tone, or implied meanings.
5. **Speech-to-Text and Text-to-Speech**
    - Converts spoken language to text and vice versa, enabling applications like virtual assistants.

Components of NLP

6. **Discourse Integration**
    - Links sentences and paragraphs to derive meaning in a broader context.
    - Example: Resolving pronouns or connecting ideas in a conversation.
7. **Machine Translation**
    - Translates text between languages using models like Google's Neural Machine Translation (GNMT).
8. **Sentiment Analysis**
    - Determines the emotional tone of text (positive, negative, or neutral).

These components work together to enable machines to understand, analyze, and generate human

Real-Time Applications in NLP

- **Chatbots and Virtual Assistants**
  - Examples: Siri, Alexa, Google Assistant.
  - Use NLP for real-time conversation, voice commands, and task automation.
- **Sentiment Analysis**
  - Analyzes customer feedback, reviews, and social media posts to assess opinions and emotions in real-time.
  - Example: Monitoring brand sentiment on Twitter.
- **Real-Time Translation**
  - Tools like Google Translate offer instant text and speech translation between languages.
  - Useful for travelers and multinational businesses.

Real-Time Applications in NLP

- **Spam Detection and Email Filtering**
  - Identifies and categorizes spam or phishing emails using NLP techniques.
  - Example: Gmail's smart filters.
- **Customer Support Automation**
  - AI-driven bots assist customers by resolving queries or escalating complex issues in real time.
  - Examples: Live chat systems on e-commerce websites.
- **Speech Recognition Systems**
  - Converts spoken words into text for applications like transcription, voice search, or dictation.
  - Examples: Otter.ai, Zoom live captions.

Real-Time Applications in NLP

- **Search Engines**
  - NLP improves the relevance of search results by understanding queries contextually.
  - Example: Google's semantic search and autocomplete.
- **Healthcare Applications**
  - Extracts patient information from medical records and supports real-time transcription during doctor-patient consultations.
  - Example: Nuance’s Dragon Medical.
- **Fraud Detection**
  - Monitors text communications and patterns for fraudulent or malicious intent in banking and cybersecurity.
  - Example: Real-time analysis of transactional data.

NLP Pipeline

An NLP pipeline is a structured workflow to process, analyze, and interpret text data. Below are the key stages:

1. **Text Acquisition**

- Collect raw text data from sources such as websites, documents, emails, or speech.
- Convert spoken language into text using **Speech-to-Text** systems if necessary.

NLP Pipeline

2. Text Preprocessing

Clean and prepare the text data for analysis.

- **Tokenization**: Split text into sentences or words.
- **Lowercasing**: Convert text to lowercase for uniformity.
- **Stopword Removal**: Eliminate common words (e.g., "is," "the") that don't add significant meaning.
- **Stemming/Lemmatization**: Reduce words to their base or root forms (e.g., "running" → "run").
- **Punctuation Removal**: Strip unnecessary symbols or special characters.

NLP Pipeline

3. Feature Extraction

•Transform text into numerical data for analysis.

- **Bag of Words (BoW)**: Represents text as a set of word frequencies.
- **TF-IDF**: Assigns weights to words based on their importance in a document.
- **Word Embeddings**: Generate dense vector representations of words (e.g., Word2Vec, GloVe).

NLP Pipeline

4. Syntactic and Semantic Analysis

•Understand grammar, structure, and meaning of the text.

- **Part-of-Speech (POS) Tagging**: Assign grammatical roles to words.
- **Dependency Parsing**: Identify relationships between words.
- **Named Entity Recognition (NER)**: Extract entities like names, places, or dates.

NLP Pipeline

5. Model Training

•Train machine learning or deep learning models on processed text data.

- Examples: Sentiment analysis, classification, machine translation, etc.
- Algorithms: Naive Bayes, Support Vector Machines (SVM), Transformer models (e.g., BERT, GPT).

NLP Pipeline

6. Evaluation and Tuning

- Assess the model's performance using metrics like accuracy, precision, recall, F1-score.
- Optimize hyperparameters to improve performance.

7. Deployment

- Integrate the NLP model into real-world applications like chatbots, search engines, or sentiment analysis systems.

NLP Pipeline

8. Feedback and Iteration

- Continuously monitor performance, gather user feedback, and retrain the model as needed to adapt to new data or requirements.

Phases of NLP

NLP involves multiple phases, each contributing to the effective processing and understanding of natural language.

1. Lexical Analysis (Tokenization)

- Breaks the text into smaller units like words or sentences (tokens).
- Removes unnecessary characters (e.g., punctuation).
- Example: Splitting "I love NLP!" into ["I", "love", "NLP"].

Phases of NLP

2. Syntactic Analysis (Parsing)

- Analyzes the grammatical structure of sentences to check if they follow syntax rules.
- Tasks:
  - **Part-of-Speech (POS) Tagging**: Identifying nouns, verbs, adjectives, etc.
  - **Parsing**: Constructing parse trees to show sentence structure.
- Example: In "He eats apples," identifying "He" as a pronoun, "eats" as a verb, and "apples" as a noun.

Phases of NLP

3. Semantic Analysis

- Focuses on extracting meaning from text.
- Tasks:
  - **Word Sense Disambiguation**: Determines the correct meaning of words in context (e.g., "bank" as a financial institution or riverbank).
  - **Named Entity Recognition (NER)**: Identifies entities like names, locations, or dates.

Phases of NLP

4. Discourse Integration

- Considers the relationships between sentences to understand the context.
- Resolves pronouns (e.g., "She" refers to a previously mentioned person).
- Example: In "Mary is a doctor. She works in a hospital," linking "She" to "Mary."

Phases of NLP

5. Pragmatic Analysis

- Interprets meaning based on situational context, user intent, and real-world knowledge.
- Example: Understanding sarcasm or implied meaning (e.g., "Great, just what I needed!" can be sarcastic).

Thank You