Tuesday, April 1, 2025 12:14 PM

Scenario

- Our brain reasons logically. we understand the language and process it accoardingly to do reasoning.

- The computer don't understand the sentences written or spoken in languages such as Nepali, or English

Facts, Rules and Ontologies

- In the sentences we speak there are some facts/assertations/axioms and we apply some logic/rules to it before speaking out something
  - We may say whether sunny so it will not rain today
  - It might not be accurate but should be sound

- As an example we discussed, interior angles of a triangle sums up to 180 degrees with the help of Euclidian geometry. However if the surface is spherical the rule does not implies.

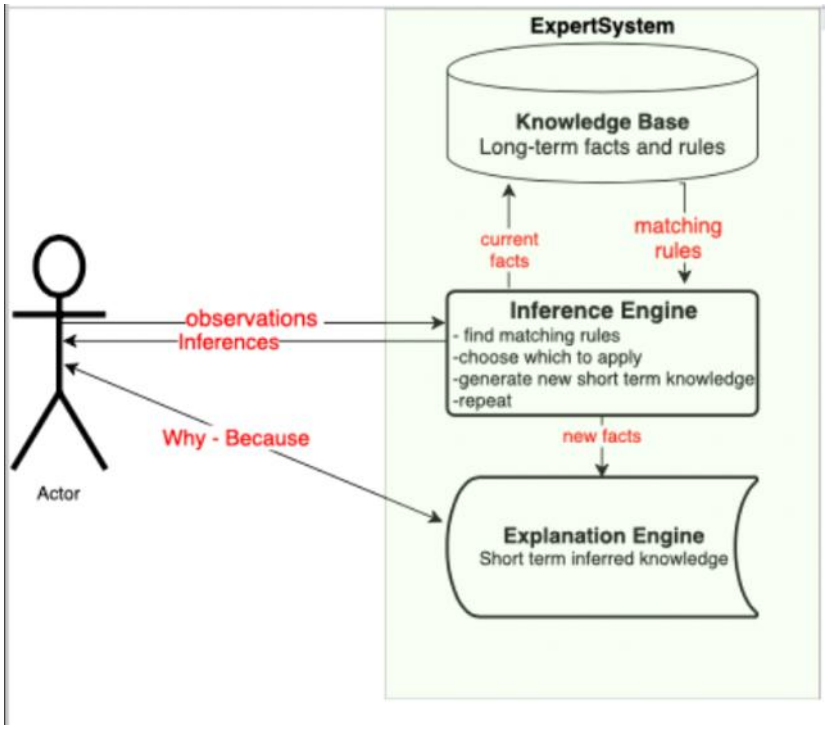
- So we are required to store information such as Objects, Events, Actions/Rules, Consequences/Results, and Meta Knowledge.

- Building blocks:
  - Facts: relation between fact and property. also called predicate and often represented as Boolean variables
    - EG: isRound(myFruit) = True
  - Rules: Asserting relationship between 2 facts. used to generate new knowledge. set of variables.
    - EG: if(isRound(myFruit)) AND isOrangeColor(myFruit) => isOrange(myFruit)
  - Ontologies: framework for organizing rules and knowledge.
    - EG: use isRound or isSpherical but not both all around
    - On websites:
      - Basis or semantic web. Structure to follow to organize data on web sites. It allows automated reasoning based on content. look schema.org which is followed by google, Microsoft and others
      - contents RDF tripplets (object, predicate, subject)

Meta Knowledge

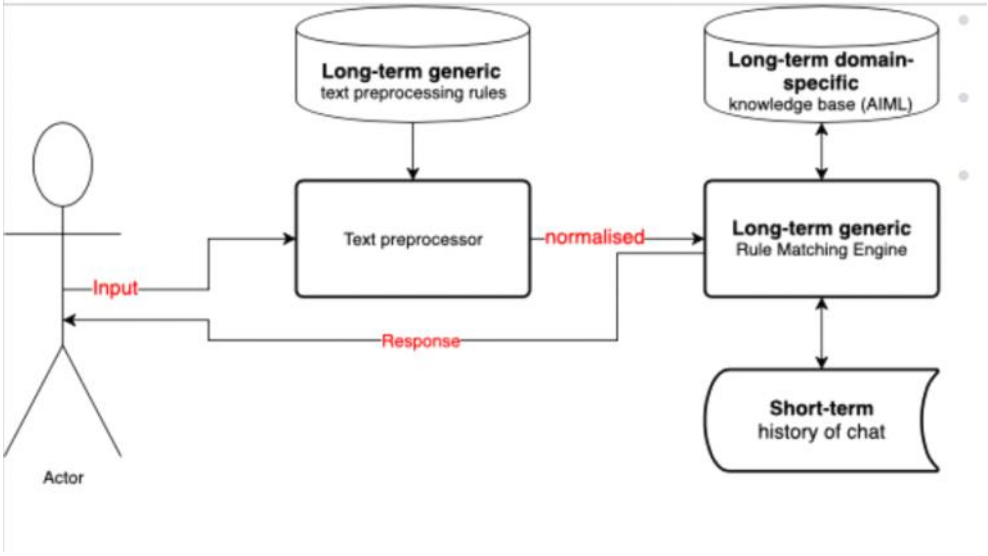
- Knowledge of knowledge
- *Explicitly Can be either*
  - Knowledge of relationship: used for reasoning
    - EG: cat belongs to mammal
    - Three components:
      - Domains: physical things on left side
        - eg: cat
      - Range: thing on the right side
        - eg: mammal
      - Cardinalities/symmetry:
        - Relationships (1 to m and …)/symmetry means can exists 0-m or vice versa relationship
        - eg: many cats to 1 mammal. 1 mammal to many cats
      - Sub class relationship:
        - 1 rule covering many facts.
        - eg: cats and dogs are mammals
      - Disjunction (logical XOR)
        - Either 1 or another not both
        - Fruit is either apple or orange not both
      - Transitive:
        - eg: cat belongs to mammal and mammal belongs to animal => cat belongs to animal
        - exception: knows is not transitive
  - Knowledge about validity (where, how long): used for storing information
- Implicitly can be
  - Long term knowledge (always true)
    - store in file and use every time while starting application
    - eg: cat is a mammal
  - Short term knowledge (true currently but might change)
    - weather is sunny
    - don't store permanently use as runtime variables
  - Generic Knowledge: (true for different domains)
    - store in place where it can be reused (websites, python libraries)
    - eg: email ids are of certain format
  - Domain Specific Knowledge: (true for some domains might not be true for others)
    - store in local/system directory.
    - eg: email ids of TBC stakeholders end with @thebritishcollege.edu.np

Expert Systems (Rule based systems):



- Consists of 3 parts
  - Knowledge bases (long term rules/facts)
    - eg: Apple is round
  - Inference Engine
    - get current facts from user and send it to knowledge base (eg: fruit is round)
    - retrieve matching rules from knowledge base
    - choose which to apply
    - generate short term knowledge
    - repeat
    - after completion sends new facts to Explanation engine
  - Explanation Engine
    - Short term generated knowledge (fruit is apple)
    - provide knowledge, explanation, reasons to user

AIML: Artificial Intelligence Markup Language: xml based language : Expert system for authoring chatbot knowledge bases



- It's made up of rules called categories
- Each category contains:
  - A pattern: the exact form of words that the chatbot recognizes to trigger the rule. i.e. condition
  - A template: the output of the rule, i.e. action
- because this is xml parts these are in the form of tags <category> </category>.
- atomic category is one having pattern and template as simple text
- can be considered as query result pair of how each results can be encoded.
- EG:

```
<aiml>
<category>
  <pattern> TELL ME ABOUT APPLES</pattern>
  <template> <srai> DESCRIBE APPLES</srai> </template>
</category>
</aiml>
```

Forward Chaining:

- Data driven discovery of knowledge
- Pseudocode
  - Initialize knowledge base with any long term facts
  - Add assertion derived from observations
  - Find match-set of rules matching condition by current short term knowledge
  - while match-set is not empty and decision is not met:
    - rank match-set using domain specific rule precedence
    - select highest ranked rule from match-set
    - apply selected rule which might let you to
      - assert new short term knowledge/asserts directly
      - ask question to get new knowledge from the user
    - Add inference to current set of shore term knowledge
    - Find the new match set
  - output result
- In aiml we can use <srai> tag pair within a templete lets you infer new things by asserting that two inputs are equivalent : if a => b and b=> c than a => c
- if input matches tell me about apples, and ***tell me about apples*** implies ***describe apples***, output respond to ***describe apples***
- In other words, ("tell me about apples" == True) => ("describe apples" == True)

Representation

- If we want to feed that information to the system we need to represent the sentences/prepositions in some kind of representation that a computer understands and put some logic/rules into it to derive hypothesis/theory/conclusion.
  - efficiency of a knowledge based agent is limited by assertions holding the true value.

- To do so we have to encode those to different forms: 1. Prepositional Logic, 2. First order predicate logic (most used)

- Symbols: negation, conjunction, disjunction, implies and biconditionals