

Artificial Intelligence

Introduction

Tools are invented to make our human life easier. For several centuries now even some decision-making has been left up to machines. For example, a machine makes the decision when to turn on an air conditioner and when to turn it off. Today, much of the complex information required to make an expert decision is provided by computers. If the information required to make a decision resides in a computer why not teach the computer to make the decision itself? Human beings make decision based on thought processes. Can machines think? Human beings learn from mistakes and adapts to new environments. Can machines learn? These are some of the questions various academic disciplines such as Computer Science, Psychology, Philosophy, Neurology and Mathematics attempt to answer through the field of study of Artificial Intelligence.

In 1997, the World Chess Champion, Garry Kasparov, played against the IBM's Deep Blue Supercomputer and lost. Does that mean that Deep Blue is more intelligent than its human opponent? Human intelligence is a controversial matter and the intelligence quotient (IQ) tests are not administered routinely anymore. Alan Turing proposed a test that would provide some insight into the issue of computer intelligence. He proposed a human judge in a secluded room to decide whether the one communicating remotely through a computer communication link with him/her is a human or a machine. Turing proposed that if, under these conditions, a judge was less than 50% accurate in properly picking, then the computer must be a passable simulation of a human being and hence, intelligent.

Artificial intelligence (AI) may be defined as a field of computer science that is concerned with mechanizing things people do that require intelligent behavior. The two most important requirements for intelligent behavior are the knowledge and searching. Thus, AI must be concerned with acquiring knowledge and application of it using rules. Domain specific knowledge can be obtained from human experts by knowledge engineers (AI programmers).

Knowledge

Data, information and knowledge carry different meanings. Data consists of raw numbers, measurements, characters, etc. without any added meaning. Information is obtained as a result of processing of the data and carries with it some specific meaning. Knowledge, on the other hand, is a collection of related facts, procedures, models and logics that can be used in learning and problem solving. Randomly picked data items from a database would not provide any meaningful results and could be classified as data. However all related data about a particular person is picked, it gives information about that person. When this information is applied to derive a solution to a problem that is not evident in the person's record, the tools and information used together is called knowledge.

Knowledge can be represented as part of a rule. Any rule consists of an ‘if part’ and a ‘then part’ as in ‘if it is cloudy then it will rain’, or ‘if Thomas is the son of John and Mercy is the wife of John then Mercy is the mother of Thomas’. Knowledge can also be represented using predicate and prepositional calculus. Examples of predicate statements are: likes(john, mercy), mother(mercy, thomas).

Searching

In searching for a solution each rule is compared with the facts contained in the database. When the ‘if’ part of the rule matches a fact, then the ‘then’ part is executed. There are two ways in which rules are compared and executed, forward chaining and backward chaining. In forward chaining (also known as data-driven search), the search begins with the given facts applying to the rules and proceeds to create new facts, which in turn may create additional new facts, eventually satisfying the ‘if’ portion. In backward chaining (also known as goal-driven reasoning), the process start with the desired goal (hypothesis) and finds rules that satisfy the goal, as in solving a maze from the finish back to the start. One of the searching methods is to create a search tree (depth-first or breadth-first) and begin searching at the root until the goal is found.

Programming Languages for AI

Programs for Artificial Intelligence can be developed using any high level language, but will require many complex ‘if then else’ statements. Common Lisp and Prolog are languages developed specifically for AI programming that reduces number of lines of code dramatically. The Lisp was developed specifically for list processing. Prolog (Programming in Logic) is used for solving problems that involve objects and relationships between objects. If a statement reads, “John is a sibling of Thomas”, it can be coded into Prolog since it describes relationship between two objects. What makes them brothers? There are various ways to describe this relationship, children have same parents, children have same father, children have same mother, etc. For example, “two children are siblings if they have the same father”, stated in Prolog:

```
sibling(Child1,Child2)
    :-father(Child1,Fname),
       father(Child2,Fname).
```

Figure 8-1 is an example of a Prolog program the author wrote. The program specifies predicates such as father(symbol, symbol), and facts (clauses) encoded in such as father(thomas,john). Prolog could read these clauses from a database rather than having hard coded. In this example, the clauses only include who the father is and who the wife is. All other relationships are derived using the rules. Program runs are given in Figures 8-2, 8-3, 8-4 and 8-5.

Predicates

father(symbol,symbol)
mother(symbol,symbol)
wife(symbol,symbol)
sibling(symbol,symbol)
grandfather(symbol,symbol)
grandchild(symbol,symbol)
cousin(symbol,symbol)
grandmother(symbol,symbol)
run(symbol)

clauses

father(thomas, john).
father(mary, john).
father(reggie, john).
father(john, avrachan).
father(mercy, thomachan).
father(avrachan, philipose).
father(biji, philip).
father(johnson, philip).
father(binu, philip).
father(philip, avrachan).
father(mathew, avrachan).
father(sam, avrachan).
father(kunjukunjamma, avrachan).
father(mary, avrachan).
father(philip, sam).
father(lennie, sam).
father(reggie, mathew).
father(shibu, mathew).
father(gracie, mathew).
father(daisie, mathew).
father(bobbie, koshy).
father(tommie, koshy).
father(saji, baby).
father(shaji, baby).
father(reggie, baby).
father(binu, thankachan).
father(bini, thankachan).
father(baby, thomachan).
wife(baby, kunjukunjamma).
wife(thankachan, baby).
wife(koshy, mary).
wife(sam, molly).
wife(mathew, madana).
wife(philip, saramma).

```

wife(john,mercy).
wife(thomachan,thankamma).
wife(avrachan,thankamma).
wife(philipose,checha).
mother(Child,Person)
  :-father(Child,X),
  wife(X,Person).
sibling(Child1,Child2)
  :-father(Child1,Fname),
  father(Child2,Fname).
grandfather(Child,Person)
  :-father(Child,Fname),
  father(Fname,Person)
  or
  mother(Child,Mname),
  father(Mname,Person).
grandmother(Child, X):-
  grandfather(Child,Y),
  wife(Y,X).
grandchild(Person,Child)
  :-father(Child,Father),
  father(Father,Person).
cousin(Child1,Child2)
  :-father(Child1,X),
  father(Child2,Y),
  sibling(X,Y)
  or
  mother(Child1,X),
  mother(Child2,Y),
  sibling(X,Y)
  or
  father(Child1,X),
  mother(Child2,Y),
  sibling(X,Y).
run(parents):-
  makewindow(1,7,7,"",0,0,25,80),
  makewindow(1,18,14,"FIND PARENTS",15,1,9,78),
  write("Enter the child's name, I WILL TELL YOU WHO THE PARENTS ARE!\n"),
  readln(Name),
  father(Name,X),mother(Name,Y),
  write(X," is the father and ",Y,
  " is the mother of ",Name).
run(mom):-
  makewindow(1,7,7,"",0,0,25,80),
  makewindow(1,18,14,"FIND MOTHER",15,1,9,78),
  write("Enter the child's name, I WILL TELL YOU WHO THE MOTHER IS!\n"),

```

```

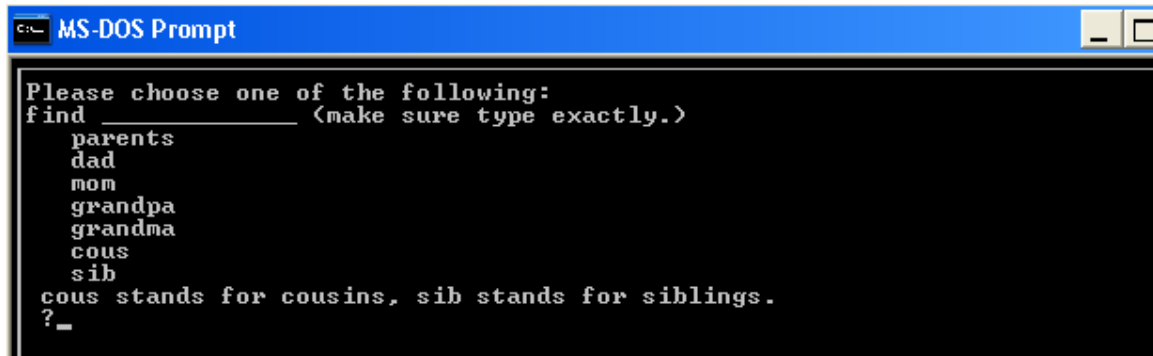
readln(Name),
mother(Name,Y),
write(Y," is the mother of ",Name),fail.
run(dad):-
makewindow(1,7,7,"",0,0,25,80),
makewindow(1,18,14,"FIND FATHER",15,1,9,78),
write("Enter the child's name, I WILL TELL YOU WHO THE FATHER IS!\n"),
readln(Name),
father(Name,X),
write(X," is the father of ",Name," \n"),fail.
run(cous):-
makewindow(1,7,7,"",0,0,25,80),
write("Enter the child's name, I WILL TELL YOU WHO THE COUSINS ARE!\n"),
readln(Name),
write(Name,"s Cousins are:\n"),
cousin(Name,X),
father(Name,Y),not(father(X,Y)),
write(X," \n"),fail.
run(sib):-
makewindow(1,7,7,"",0,0,25,80),
makewindow(1,18,14,"FIND BROTHERS AND SISTERS",15,1,9,78),
write("Enter the child's name, I WILL TELL YOU WHO BROTHERS AND
SISTERS ARE!\n"),
readln(Name),
sibling(Name,X),
write(X," is the sibling of ",Name," \n"),fail.
run(grandpa):-
makewindow(1,7,7,"",0,0,25,80),
makewindow(1,18,14,"FIND GRAND-FATHER",15,1,9,78),
write("Enter the child's name, I WILL TELL YOU WHO THE GRAND-FATHER
IS!\n"),
readln(Name),
grandfather(Name,X),
write(X," is the grand father of ",Name," \n"),fail.

run(grandma):-
makewindow(1,7,7,"",0,0,25,80),
makewindow(1,18,14,"FIND GRAND-MOTHER",15,1,9,78),
write("Enter the child's name, I WILL TELL YOU WHO THE GRAND-MOTHER
IS!\n"),
readln(Name),
grandmother(Name,X),
write(X," is the grand mother of ",Name),fail.

goal

```

```
makewindow(1,7,7,"",0,0,25,80),
write ("Please choose one of the following:\n"),
write ("find _____ (make sure type exactly.)\n"),
write (" parents \n"),
write (" dad\n"),
write (" mom\n"),
write (" grandpa \n"),
write (" grandma\n"),
write (" cous\n"),
write (" sib\n"),
write (" cous stands for cousins, sib stands for siblings.\n"),
write (" ?"),
readln(Choice),
run(Choice).
```



```
MS-DOS Prompt
Please choose one of the following:
find _____ (make sure type exactly.)
  parents
  dad
  mom
  grandpa
  grandma
  cous
  sib
cous stands for cousins, sib stands for siblings.
?_
```



```
FIND FATHER
Enter the child's name, I WILL TELL YOU WHO THE FATHER IS?
thomas
john is the father of thomas
Press the SPACE bar
```

```
Enter the child's name, I WILL TELL YOU WHO THE COUSINS ARE!
thomas
thomas's Cousins are:
hiji
johnson
binu
philip
lennie
shibu
gracie
daisie
binu
bini
hobbie
tommie
saji
shaji

Press the SPACE bar
```

```
FIND GRAND-FATHER
Enter the child's name, I WILL TELL YOU WHO THE GRAND-FATHER IS!
thomas
avrachan is the grand father of thomas
thonachan is the grand father of thomas

Press the SPACE bar
```

Application of Artificial Intelligence

Applications of AI can be categorized into game playing, speech recognition, natural language processing, automated planning and scheduling, autonomous control, robotics, computer vision and expert systems.

Human beings have incorporated game playing into their routines for as long as civilizations existed. for entertainment as well as for settling disagreements. Chess was played by kings as an alternative to sacrificing their subjects. Chess playing has become the ultimate challenge to AI.

Natural language processing is perhaps the most challenging problem for AI. It involves parsing the sentence, recognizing the context, and current societal issues. For example, 'cool' means different things at different times. Natural language processing is limited extremely narrow domain specific areas. Speech recognition has gained grounds. Programs can recognize words and display it with more than 90% accuracy. However, it requires extensive training.

- Planning and scheduling
- Autonomous control
- Robotics
- Computer vision

Expert Systems

Neural Network

Fuzzy Logic