International Journal of Scientific Research & Engineering Trends
Volume 8, Issue 2, Mar-Apr-2022, ISSN (Online): 2395-566X

Introduction to Artificial Intelligence and Problem Solving

Kartik Sharma
A High school student,
Kendriya Vidyalaya NFC,
Vighan Vihar Delhi- 110092, India

Yash Bhargav
Computer Science Engineering,
Dronacharya Group of Institutions,
Greater Noida

Abstract- Artificial intelligence is the field devoted to building artifacts capable of displaying in the controller, a well-understood environment, and over a sustained period behavior that they consider being intelligent or more generally, behavior that we take to be at the heart of what it have to a mind. In this paper we discuss about one of the upcoming fields in artificial intelligence and problems solving which is automatic speech recognition from neural signals.

Keywords- Artificial intelligence, Problem Solving, AI technique, Algorithm.

I. INTRODUCTION

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by animals including humans. Artificial intelligence problems span a very broad spectrum. They appear to hear very little in commonly expected that they are hard to get techniques that are appropriate for the solutions of a variety of these problems.

II. METHODOLOGY

Q1. What is artificial intelligence?
Artificial intelligence is the field devoted to building artifacts capable of displaying in the controller, a well-understood environment, and over a sustained period, behavior that they consider being intelligent or more generally, behavior that we take to be at the heart of what it have to a mind. This is itself a deep philosophical question and attempts to systematically answer it falls within the foundation of artificial intelligence as a rich topic for analysis and debate. Nonetheless, a provisional can be a given answer.

Q2. How do humans manage to behave intelligently?
This question is empirically, it's predominantly for psychology and cognitive science to answer. However, this question is pertinent this is because any insight into humans might help build machines that work similarly.

Q3. What is to have a mind?
This question on what is the mark on mental is philosophical. Thrust on artificial intelligence has left significant urgency to it. Carefully philosophical contemplation of this question has influenced the topic of artificial intelligence itself.

3.1 Foundation of AI and ML: Supervised, unsupervised learning, reinforcement learning, training/testing database, trained model, optimization algorithms and machine learning pipeline.

3.2 Dimension of Artificial Intelligence:

- **Think like a human-** Model human cognition. The 1960s "cognitive revolution" information-processing psychology.
- **Think rationally-** Formalize the inference process. Several geek schools developed various forms of logic. Notation and rules of derivation for thoughts may or may have not proceeded to this idea of mechanization. The main obstacle to the logistic approach in building programming to crate intelligence. Not all intelligence behavior is mediated by logical deliberation. Inform knowledge is not precise; difficult to model uncertainty. Theories and practices are hard to put together.
- **Act rationally-** Doing the right thing. One which is expected to maximize the goal achievement; given the available information. Does not necessarily involve thinking examples - bionic reflex more general logic approach. Amendable to scientific development than the approach based on human behavior or human thoughts. Achieving perfect rationality in a couple of environments is not possible because the computational demand is too high.
- **Act like a human-** Exhibit human behaviors. Crates machines that perform functions that required intelligence when some task is performed by people.

3.3 Several capabilities need to be incorporated:
Natural language processing, knowledge representation, automated reasoning, machine learning, computer vision, machine learning, deep learning, neural network and robotics etc.

Q4. Can machines think?
Alan turning, laying the ground for what was later known as artificial intelligence starts his landmark paper .Computing machinery and intelligence with the word.

Q5. Early days of AI?
In later 1955, Allen Newall and Herbert Simon developed. The logic theorist is considered by many to be the first AI program. The programmers representing each problem as a tree model would attempt to solve it by selecting the branch that would most likely result in the correct conclusion

Q6. Weak V/S Strong AI?
Weak AI aiming up to build machines that act intelligently without mathematics taking a position on wherever or not the machines are intelligent. Strong AI is the field devoted to building person Charniak and McDermott (19850 concerned is their classic introduction to AI that are very far from achieving strong AI. The ultimate goal of AI wishes we are very far from achieving it builds to people or more humbly on animals. The fundamental goal of AI research is not merely to mimic intelligence or produce some clever fake technology.

Not at all AI wants only the genuine article: machines with a mind in the full and literal sense. This is not science fiction, but real science, based on the theoretical conceptions as deep as its daring. Namely, we are not rooted, computers ourselves. The goal of work in artificial intelligence is to build machines that perform tasks normally requiring human intelligence or working better than human and also consuming less time as well.

Q7. What is involved?
Interaction with real-world examples perceives understanding and acting. For example

- Speech recognition
- Image understanding
- Reasoning and Planning involving
- Modeling external world

Planning and decision making and dealing with unexpected problems and uncertainty. Learning and adaptation thought internal model being always updated such as baby learning to categorize and recognized animals

Q8. What is involved?

- **Philosophy -** Logic a method for reasoning, mind as a physical system. Foundation of learning/ language rationality
- **Mathematic -** Formal representation and proof theory. The algorithm, computation decidability/ traceability.
- **Statistics / Probability -** Modeling uncertainty, learning from data.
- **Economics -** Utility, decision theory.
- **Neuroscience-** Neurons as information processing units.
- **Psychology/Cognitive science-** how people behave and perceive the process. Cognitive information, represent knowledge.
- **Computer engineering** – Building a fast computer
- **Control theory-** Designing systems that maximize objective function over time.
- **Linguistics-** Knowledge representation, grammar

8.1 Production system:
Components of a production system, basis procedure, Irrevocable, Back training, Graph search, Problem representation, specialized production system, commutative production system, decomposable production system.

Q9. What is an AI technique?
Artificial intelligence problems span a very broad spectrum. They appear to hear very little in commonly expected that they are hard to get techniques that are appropriate for the solutions of a variety of these problems. One of them for hard and fast results to come out of the first 3 decades of AI research is the realization that intelligence required knowledge.

AI technique is a method that exploiters knowledge and should be represented in such away.
Knowledge captures generalization

- Understood by people who provide it easily be modified.
- Use to help overcome, its sheer bulk. To build a system to solve a particular problem, we need to do four basic things - Define the problem precisely, analyzing the problem, Isolated and represent task knowledgeable and choose the best problem-solving techniques.

Q10. What is AI technique?

- **Traditional programming** – Programs can answer only the specific questions it's meant to solve. Modification is program deeds to change its structure. Modification may deed to affecting the program adversely.
- **Programming with AI** – Programming with AI can answer generic questions it's designed to solve. Modification ion programs that don't change the structure of independent pieces inform. Quick and easy program to modify.

III. MODELING AND ANALYSIS

1. The problem as a state-space search:
A problem-solving task can be formulated as a search in state space. A state-space consists of all the states of the domain and a set of operators that change one state into another. The states can be best thought of them as nodes in a connecting graph and operators as edge.

Certain nodes are designated as goal nodes and a problem is said to solve in a path from an initial state of a goal state have been found. The set of all certain configurations of relevant objects in the space of the problem state or the problem space. This is also called state space.

Play chess by starting at an initial state, using a set of rules to move from one state to another, and attempting (search) to one of a set of final states. The state's space representation seems natural for chess because the set of states corresponds to a set of a board positions. This kind of representation is also useful for less structured problems with the use of a more complex structure than a matrix.

2. A formal description of the problem:

- **States space-** Define a state space that contains all the possible configurations or a relevant object.
- **Initial space-** Specific one more state within the spaces as possible situations from which problem-solving can start.
- **Goal states -** Specific one or more states that would be acceptable as a solution to the problem.
- **Operations-** Specific a set of rules that describe the action (operators) available.

3. Production system and AI:

A system consisting of separate, database, operations and manage components represents the ideal metaphorical building block for constructing lucid descriptions of AI systems. AI systems display a more or less rigid separation between computational components of data, operations, and control. Various generalization of the computational formalism known as the production system devote a clean separation of these computational components and thus seems to capture the essence of operations of the money AI system.

Choosing rules and keeping tracking of those sequences of rules already tried to contribute to the AI production system can thus characterize the search process in which rules are tried until it sequence of them is found that produces a database satisfying the termination condition.

4. To solve a problem statement into these 3 components of a production system is called the representation problem in AI:

Global Database, Rules, Control Strategies Transforming the problem statement into these 3 components of a production system is called the representation problem in AI. A production system consists of a collection of production (rules), working memory of facts (Database), and an algorithm for producing new facts from also.

4.1 Production system v/s Conventional computations:

There are several differences between production system structure and conventional computational systems that use hierarchically organized programs. The production system structure is modular changes is any of the components can be made independently. Using a conventional computation in AI application is difficult, for any change in knowledge base would require extensive changes.

4.1.1Production System: States description of the global database (F-Rule) Forward production system. Global description as the global database (B-Rule) Backward production system

4.1.2Specialized production system: Commutative production system order in which a set of applicable rules is applied to a database is unimportant.

4.1.3Decomposable production system: Initial database or split into separate components that can be processed independently.

4.1.4Commutative production system: A production system is commutative if it has the following properties for the database each set of rules applicable to D is also applicable to any database produced by applying an applicable rule to D. Commutative production systems are an important subclass that has the following properties .An irrevocable control regives can always be used in a commutative system because the application of the rule never needs to be taken back or undone. No need for a mechanism for applying the alternative sequence of rules. Rules that apply to an earlier DB apply to the current one. Applying inappropriate rules delays but never prevents termination.

4.1.5Decomposable production system: Consider a system- the initial database is (C, B, Z). The production rule is based on following rewrite rules

- R1: C (D, L)
- R2: C (B, M)
- R3: B (M, M)
- R4: Z (B, B, M)

The termination condition is that the database condition only MS. A graph search control strategy might explore many equivalent paths in producing a database condition only MS.

Redundant paths can lead to inefficiencies because the control strategies might attempt to explore all of them, worse it might do work that is wasted ultimately in exploring paths that don't terminate. One way to avoid the exploration of these redundant paths is to recognize that the initial database can be decomposed or split into separate components that can be processed independently.

4.2 AND-OR Graph:

Nodes labeled by the components database have sets of successor nodes each labeled by one of the components. These nodes are called nodes because to process the compounds database to termination the entire components database must be processed by termination.

A structure called AND-OR GRAPHS is useful for depicting the activity of the production system under the control regive of a decomposable production system. The notion of a decomposable production system encompasses a technique often called problem reduction in AI. Problem reduction idea usually involved replacing a problem goal

International Journal of Scientific Research & Engineering Trends
Volume 8, Issue 2, Mar-Apr-2022, ISSN (Online): 2395-566X

with a set of sub-goals such that if the sub-goals are the main goal is also involved. Explaining the terms of the decomposable production system allows us to be indefinite about whatever we are decomposing problem goals or problem states.

4.2.1 Knowledge of problem domain: An efficient AI system required knowledge of the problem domain. This knowledge can be subdivided into 3 board domain categories- Declarative knowledge, Procedure knowledge, and Control knowledge.

4.2.2 Uniformed Search: Review problem solving as a search. State spaces, Graph search, Generic searching algorithm.

4.2.3 Uninformed search strategies: Problem-solving as states space search. Can be abstracted to the mathematical hassle of finding a path from the begin node to a goal node in a directed graph.

4.2.4 Graph: Basic definition: Graph G comprises the set V of verticals and sets E of edges. A vertex set can be anything, but it is most commonly a collection of letters or numbers. A set of edges is a doubleton subset of V.

A, B ⍰ V, and A≠B

If G (V, E) is a graph and {A, B}. E, then we say vertices a and b are adjacent and edge {a, b} joins them for connects them or is incident on them. We call a and b the endpoint of the edge. Two edges that share one vertex, such as {a, b} and {b, c} with a ≠ c are adjacent to each other.

4.2.5 Graph: Paths: A notice of path in a graph is intuitively clear but a little hand to pin down formally. Suppose G (V, E) is a graph with vertices and edges. An alternating sequence of verticals and edges is a path from length.

4.2.6 Directed Graph (Digraphs): A directed graph (digraph, for short) G consists of a set V of vertices and a set E of directed edges. A directed edge is an outer pair of elements of V. put another way, the pain G (V, E) with E v × v is a digraph. Digraphs allow for lops of the forms (a, a). It is also possible to have two edges (a, b and (b, a) between verticals a and b. We make ordering the pair in E to give each edge of direction; namely, the edge (a, b) goes from a to b. when we draw the digraph, we draw the edge (a, b) as an arrow a to b.

4.2.7 Graph searching: A problem of finding a sequence of actions to achieve a goal is abstracted as searching for a path in a directed graph. To solve a problem, define the underlying search space and apply a search algorithm to that search space. Searching in a graph provides an appropriate abstract model of problem-solving independent of a particular domain (Formalizing search in state-space). Each model is a data structure that contains a state description plus other information such as the parent of the node, the operation that generates the node from the parent, and other bookkeeping data. Each and every arc corresponds to an instance of one of the operators. A state space is a graph (V, E) achieve. V is a set of nodes and E is a set of arcs and each arc is directed from one node to another node. Each arc has a fixed positive cast associated with it corresponding to the cost of the operator. Each node has a set of successor nodes corresponding to all of the legal operators that can be applied at the source node state. One or more nodes are designated as start nodes. A intention test predicated is applied to the state to determine if its related node is a goal node. The cost of a solution is the sum of the arc cost of the solution. State-space is searched is a process of making explicit a sufficient portion imp-licit state-space graph to find the goal node.

4.2.8 Evaluation search strategies:

- **Completeness -** Guarantees to find a solution where it exists.
- **Time complexity -** How long (worst or average case) does it find a solution?
- **Space complexity -** How much space is used for algorithms? Usually measured in terms of the maximum size of the nodes list in some stage of search.
- **Optimality/Admissibility -** If a solution is found, is guaranteed to be an optimal one?

4.2.9 Uniformed V/S Informed searches:

- **Uniformed search strategies**: Also known as "blind search" use no information about the likely direction of the goal nodes (S).
- **Uniformed search method**: Breadth-first, depth-first depth limited uniform cost depth-first interactive depending bidirectional.
- **Informed search strategies**: Also known as "Heuristic search" informed search strategies "use information about the domain" head in the general direction of goal nodes (S). Informed search method: hill-climbing, best first, greedy search.

4.2.10 Types of Heuristic functions: Heuristic, Heuristic function, Heuristic, and AI, Problem relaxation, Admissible Heuristic, Composite Heuristic, Consistent Heuristic

4.2.11 Heuristic -The study of the method and rules of discovery and invention. Used are knowledge of the problem to consider (some not all) successor of the current state (preferably just one< as with an oracle). This means pursuing the state space, gaining, speed, built perhaps missing the solution.

4.2.12 Heuristic function: For a heuristic search to work, we must be able to rank the cildren to a node. A heuristics function takes a state and returns a numeric value- A composite assessment of this state.

4.2.13 Heuristic and AI search: In AI search, the word heuristics has a very specific meaning. A heuristic is a function that estimates how close a state in a goal.

The heuristics function takes a state and returns and estimates its distance (path cost) from a goal state.

4.2.14 Problem relaxation: A standard approach to creating heuristics is problem relaxation we relax a problem by adding some new actions such that a search is not required to find the solution cost in relaxation problem. Heuristic created as a solution to relaxed problems is usually admissible because adding new actions should only reduce the solution cost and node increases it.

4.2.15 Admissible heuristics: A heuristics is admissible if it in no way overestimates the cost of reaching a purpose from any node. The only thing required for a heuristics to be admissible is that it never returns as a value greater than the actual path cost to the nearest goal from any node.

4.2.16 Composite heuristic: The maximum of two admissible heuristics is also admissible If we have developed two or move heuristics for the same problem and are ensure off whether any of them dominates all others, we can use the maximum of them because of the composite heuristic. This composite heuristic will take more time to compute but however would be more accurate.

4.2.17 Consistent heuristic: Consistent heuristic placed even stricter constraints on the heuristic. It required that the estimate must never be greater than the actual cost for each arc along a path to a goal. Consistency implies admissibility example - A consistent heuristic is also admissible. Although most admissible heuristic is consistent especially for relaxed problems.

4.2.18 Heuristic-A-Trade-Off: The heuristic has a trade-off between quality of estimate and work per node. A heuristic receives in the direction of the true value; we make bigger fewer nodes but usually do more work in line with node to compute the heuristics itself. At the two extremes are null heuristics (which always return 0 and required the least amount of work) and the exact cost.

4.2.19 Choosing a Good heuristics: For a given problem there might be many different heuristics one can choose. However, some heuristics are better than others. We can say that heuristics is better if its needs a few nodes to examine in the search tree. We also call this kind of heuristic better informed. Efficiency is very important in picking heuristics.

4.2.20 Informed search: Uniform search method, whatever breadth-first or depth-first one exhausting method for finding a path to a goal node. Informed search methods use task-dependent information to help reduce the search. Interested in minimizing some combination of the cost of the path and the cost of search required to obtain the path. Heuristics information is used to that search expands along with those sectors the from tier through to be most promising. Also called heuristic search in a heuristics search each state is assigned a heuristic value that the search in selecting the best next step. A heuristics is an operationally effectively nugget of information on how to direction search in a problem space.

4.2.21 Heuristic function: The heuristic function, h (n); estimates the cost of a goal most promising states is selected. For ordering nodes for expansion, we need a method to compute the promise of a node. This is done using a real-value evaluation function.

4.2.22 Best First Search: In a way of combing the advantages of both depth-first and breadth-first search into a single method. Idea: use an evaluation function f (n) for each node. Implementation: order the nodes in the fridge in decreasing the ordering of desirability.

4.2.23 Algorithm A: Idea: avoid expanding paths that are already expensive. Evaluation Function_ can be defined so that its value at any node estimates the sum of the cost of minimal cost from start nodes S to the nods together with the cost of minimal cost path from node n to the goal node. Let function K (n, nj) given the actual cost a minimal cost path between two arbitrator nodes ni and nj. H(n) is the cost of the minimal cost path from n to a goal node. The cost of an optimal path from a given path node n is K (s, n). The graph search algorithm using the evaluation function for ordering nodes is called algorithm A.

4.2.24 Observation: G last us choose which nodes to expand next on the basis not only of how good the node itself looks (as measured by h) but also based on how good the path to be node was an estimate of h distance of a node to the goal: if a perfect estimate of h A will converge immediately to a good with no search.

4.2.25 Local search algorithm: In many optimization problems, the state space is the space of all state space of all possible complete solutions. These algorithms don't systematically explore all the state space.

4.2.26 Local search: Using a single current state as a move to neighboring states. An idea starts with an initial guess at a solution and increases mentally until it's one.

4.2.27 Advantages: Use very little memory: Useful for pure optimization problem: Find or approximately best state according to some objective functions.

4.2.28 Constraint satisfaction problem: In state search, we explore the video idea that problems can be solved by searching in a space of states. Constant satisfaction problem examines problems< whose states and goal test conform to the standard and structured representing. A state of problem is denied by an assignment of value to some or all the variables. A method to CSP is a completem project that satisfies all of the constraints.

4.2.29 Standard search formation: Let's start with a basic approach and then improve it.

4.2.30 Notes – this is the same for all CSPs. every solution appears at depth n with n variable. The path is irreverent so can also we complete state formation. Domain of size d, factor of branching b = (n-l) d at depth l.

4.2.31 Back training search: CPS can be solved by a specialization version of depth-first search.

4.2.32 Key intuitions- we can build up to a solution by searching through the space of partial assignment. The order in which we assign the variable doesn't matter eventually they all have to be assigned. If during, we immediately reject all possible ways of extending the current partial assignment. Backtracking search is the fundamental uninformed algorithm for solving CSPs. Intern backtracking search is useful for depth. First search cost for one variable at a time and backtracking when no legal value is left to assign. Plain backtracking is an uninformed algorithm. Don't expect to be very effective for large problems.

4.2.33 Improving backtracking search: We can solve CPSs effectively without such domain-specific knowledge general-purpose method that addresses the following questions-

- Which variable have to be assigned subsequent, and in what order have to its value be attempted?
- All the implications of the current/current variable assignments for the opposite unassigned variables?
- **A general-**Purpose search method can be a huge gain in speed
- **Ordering-** In what order the value be tried?
- **Filtering -** Can we discover inevitable failure early?
- **Structure -** Can we take advantage of the problem structure?

4.2.34 Variable and value ordering:

- Minimum remaining value (MRV)
- Choose the variable with the fewest legal value. By defaulting selecting the unassigned variable simplifies selecting the next unassigned variable in the order given by the list variable.

4.2.35 Degree heuristics - choose the variable from the most constraint on remaining vars. It attends to reduce the branching factor on future choices by selecting the variable that is involved in the largest one of constraints on other unassigned variables.

4.2.36 Least containing value - Choose the one out that the few values remaining variable. Once a variable is selected the algorithm must decide on the order in which to examine its value if prefer the value that rules out the fewest choices of neighboring variable in the constraint graph.

IV. RESULTS AND DISCUSSION

AI technique is a method that exploiters knowledge and should be represented in such away, types of Heuristic functions, local search algorithm, production system and conventional computations. Algorithm A and Evaluation search strategies like Completeness, Time complexity, Space complexity. Optimality/Admissibility. The ultimate goal of AI wishes we are very far from achieving it builds to people or more humbly on animals.

V. CONCLUSION

Artificial intelligences are designed to learn on their own and resemble a human brain and physical and mental properties. One thing is for sure, is that artificial intelligence will continue to develop because of humans. Humans will continue to make new discoveries and discover new things.

Artificial intelligence will never be able to accomplish that; however they may assist a human by providing theories. The future is unknown and maybe artificial intelligence and humans will be able to work together on many different topics.

REFERENCES

[1] Newell, A. and Simon, H. A. , GPS: A Program that Simulates Human Thought, in Computers and Thought, Feigenbaum, E. A. and Feldman, J. (eds.), McGraw-Hill, New York, NY, pp. 279–293, 1963. Google Scholar

[2] Hamet, P, Tremblay, J. Artificial intelligence in medicine. Metabolism 2017; 69s: S36–S40. Google Scholar | Crossref | Medline.

[3] https://computerscienceai.wordpress.com/conclusion/#:~:text=Conclusion%20In%20conclusion%2C%20artificial%20intelligence%20will%20become%20more,ca