Name : Mohsin Karovaliya, Unity Id: mrkarova, Student Id: 200255290

ARTIFICIAL INTELLIGENCE

Assignment I

1. [20 pts.] Suppose we have an agent who works for the Hunt Library. The agent is assigned a task of transferring a set of boxes one-by-one by lifting them from location A and placing them in location B inside the building. Sensors can tell it whether the agent is near its destination or not. The rooms have stationary obstacles whose locations are not initially known. If the agent bumps into an obstacle, it will drop the box it is carrying. Some boxes contain fragile contents which will break if dropped. The environment contains obstacle-free paths of various lengths. Answer the following questions about the agent.

1. Define a PEAS specification for the agent.

   Following are the PEAS specifications for the agent:

   Performance Measure (P) :

   a) Avoid Obstacles

   b) Reach Destination

   c) Minimum Distance Covered

   d) Safety

   e) Transfer the boxes

   Environment (E) :

   a) Boxes

   b) Rooms

   c) Obstacles

   d) Freeways

   Actuators / Actions (A) :

   a) Move forward or backward

   b) Wheel / Rotate

   c) Pickup and Place boxes

   d) Pedals / Accelerate and brake

   e) Robotic arms

   Sensors (S) :

   a) GPS

   b) Accelerometer

   c) Gauges

   d) Camera / Video

2. Is it sufficient for the agent to be simple reflex? Why or why not? And if not what level is necessary?

No, it is not sufficient for the agent to be a simple reflex. In this scenario, even though the agent just needs to follow simple set of rules, it need to keep track of the obstacles. As the agent does not know about the obstacles location initially, it needs to learn and remember over the period of time. In order to support this Model-Based Reflex Agent could be helpful since it maintains the internal representation of the external world and learns about the environment over the period of time. This could help agent to become more efficient as it continues to learn.

3. Would the ability to move randomly help agent performance or not? Identify possible disadvantages to this sort of movement.

   No, the ability to move randomly won't help agent performance. Due to randomness, the agent may try to move the direction which could possibly contain more previously faced obstacles. Hence, randomness in the movement would not allow the agent to improve and take heuristic decisions.

4. Suggest one improvement to the agent design. Since every improvement carries drawbacks, what are the drawbacks to yours?

   One improvement in the agent design could be the ability to accelerate and decelerate. Ability to decelerate when obstacle is visible could help prevent agent from bumping into the obstacles and breaking objects. Deceleration could increase the time while traversing.

2. [20 pts.] Fill in the following table with proper description of the agent's environment.

	Fully Vs Partially Observable	Deterministic Vs Stochastic	Episodic Vs Sequential	Static Vs Dynamic	Discrete Vs Continuous	Multiagent Vs Single Agent
Vacuum Cleaner Agent	Partially Observable - Since it need to clean only the locally visible surface areas and need not worry about distant areas.	Deterministic - Since agent does know if the already the surface area gets cleansed after vacuuming	Sequential - Since the agent must not vacuum already vacuumed surface. This would prevent agent from doing unnecessary work.	Static - Since the environment surrounding the agent does not change by external factors if inside a closed room.	Discrete - Since the vacuum cleaner can only move n specific pre-programmed directions.	Single Agent - Since there are no external agents.
Google Car	Partially Observable - Since It	Stochastic - Since the state of	Sequential - Since the direction	Dynamic - Since the environme	Continuous - Since the car is not	Multi agent - Since there can

	only need to view the nearby road.	environment does not change based on action taken.	to move on depends on when the direction was changes previously by it.	nt changes by multiple cars.	confined by any set of points or movements.	be multiple cars present on the road.
Search and Rescue Robots	Partially Observable - Since robot need require to know entire environment at the beginning but discover later on.	Stochastic - Since it does not matter for robot what was the previous state.	Sequential - Inorder to reach the rescue scenes, it needs to search and gradually reach the location. Also rescuing requires certain set of procedures.	Dynamic - Since between state of the environment changes when robot takes actions to rescue.	Continuous - Since the robot is not confined by any set of points or movements.	Multi agent - Since there can be multiple agents in the environment.
Document Categorizer	Partially Observable - Since it just need to view the single document to be categorized at any single moment.	Deterministic - Since it knows that the category in which the document is assigned to is correct.	Episodic - Since it does not matter what was the previous state of the agent.	Static - Since there are no external factors influencing agent's procedure.	Discrete - Since the input and output location of the document is fixed.	Single Agent - Since there are no other agents.

3. [10 pts.] Answer the following questions.

1. Describe a PEAS (R&N Ch. 2) specification for Watson.

   Performance Measure:

   Buzzing before other players

   Giving correct answers to questions

   Giving answers to questions before time runs out

   Environment:

   Host of the game

   Human Competitors

   Computer asking questions

   Audiences

   Actions / Actuators:

   Buzzing the buzzer

   Text to voice conversion

   Sensors:

   Listening to questions

   Voice to text translation

   Identifying the type of question

2. Describe Watson's environment (Full/Partially Observable, Deterministic/Stochastic etc.).

Following is the Watson's environment:

Partially Observable:

Since at a give point of time, Watson just knows category of questions to be asked, and not the question itself or other categories remaining.

Deterministic:

Since Watson, knows the confidence levels of each answer to the question. This confidence level is directly related the determinism of the correctness of the answer.

Sequential:

Since Watson is aware about the state of the game at any given point. For example, it knows the category of the question being asked or the score it has received by far in the game.

Dynamic

Since the state of environment changes as and when game progresses. For example, the score changes, the categories of questions change, etc.

Continuous

Since there can be infinite questions asked in the game, and approach to find answers is different for each question.

Multiagent

Since there are other agents involve in the game (environment) which change the state (score) of the environment as and when game progresses.

3. Discuss at least three separate aspects of the Jeopardy problem domain together with the hardware and/or software design choices in Watson that are rational given those problem aspects.

Following are three aspects of the Jeopardy problem:

The Categories

In Jeopardy there are total five clues each of which have different categories. Categories can be like history, science, politics, etc. Depending on the categories and clues, Watson identifies the answer for the question.

The Questions

The questions can be of one of the following types: Decomposition and Puzzles.

Watson must first be able to identify the type of question in the clue. After identifying the type, it must parse the question differently based on these types. For example, in decomposition, Watson must break the clue into two or more sub-types. Then find the answers to sub-types to get the final answer. In puzzle, Watson must first be able to identify the type of puzzle from types like, before-after, rhyme, sounds like, math problems, etc. After identifying the type of puzzle, it must find answer to the clue.

The Domain

In this case, Lexical answer type is used to find the answer from the clue. The clues can indicate lexical answer type directly or indirectly.

4. Describe the DeepQA approach developed for Jeopardy and name the six architectural roles that are designed in this model.

DeepQA is a massively parallel probabilistic evidence-based architecture having principles such as many experts, pervasive confidence estimation, and integration of shallow and deep knowledge. For Jeopardy, more than 100 different techniques were used for analysing natural language, identifying sources, finding and generating hypotheses, finding and scoring evidence, and merging and ranking hypotheses. All these techniques were combined to improve accuracy, confidence and speed. [1] Following are architectural roles that are designed in this model:

Content Acquisition

Question Analysis

Hypothesis Generation

Soft Filtering

Hypothesis and Evidence Scoring

Final Merging and Ranking

4.

1. In following case the path from A to B is different from that of B to A. Also the number of explored nodes are different in both cases:

A*

explored nodes : keyWest, tampa, lakeCity, orlando, daytonaBeach, tallahassee, jacksonville, albanyGA, savannah, augusta, macon, atlanta, chattanooga, pensacola, nashville, charlotte, greensboro, westPalmBeach, raleigh, miami, newOrleans, memphis, batonRouge, lafayette, norfolk, richmond, littleRock, washington, beaumont, baltimore, houston, philadelphia, mexia, dallas, pittsburgh, tulsa, cleveland, newYork, austin, columbus, dayton, kansasCity, stLouis, cincinnati
total explored nodes : 44
nodes in path : keyWest, tampa, lakeCity, tallahassee, albanyGA, macon, atlanta, chattanooga, nashville, memphis, littleRock, tulsa, kansasCity, stLouis, indianapolis
total nodes in path :15
total distance : 2647.0

explored nodes : indianapolis, cincinnati, dayton, columbus, stLouis, cleveland, pittsburgh, kansasCity, buffalo, tulsa, rochester, littleRock, wichita, toronto, philadelphia, baltimore, washington, richmond, memphis, oklahomaCity, norfolk, albanyNY, newYork, ftWorth, raleigh, nashville, greensboro, chattanooga, atlanta, charlotte, macon, augusta, albanyGA, boston, providence, savannah, tallahassee, jacksonville, newHaven, daytonaBeach, orlando, lincoln, lakeCity, ottawa, tampa, stamford, omaha, westPalmBeach, miami, desMoines, montreal, pensacola, saultSteMarie
total explored nodes : 53
nodes in path : indianapolis, cincinnati, dayton, columbus, cleveland, pittsburgh, philadelphia, baltimore, washington, richmond, norfolk, raleigh, greensboro, charlotte, augusta, savannah, jacksonville, daytonaBeach, orlando, tampa, keyWest
total nodes in path :21
total distance : 2647.0

3. Following is the path wherein the path from A to B is longer in Greedy as compared to A* algorithm.

Greedy

explored nodes : miami, westPalmBeach, orlando, tampa, lakeCity, tallahassee, pensacola, newOrleans, batonRouge, lafayette, beaumont, houston, mexia, dallas, denver, grandJunction, provo, coloradoSprings, santaFe, albuquerque, elPaso, tucson, phoenix, yuma, sanDiego, losAngeles, sanLuisObispo, salinas, sanJose, oakland, sanFrancisco, sacramento, reno, pointReyes, redding, medford, eugene, salem, portland, seattle

total explored nodes : 40
nodes in path : miami, westPalmBeach, orlando, tampa, lakeCity, tallahassee, pensacola, newOrleans, batonRouge, lafayette, beaumont, houston, mexia, dallas, denver, coloradoSprings, santaFe, albuquerque, elPaso, tucson, phoenix, yuma, sanDiego, losAngeles, sanLuisObispo, salinas, sanJose, oakland, sanFrancisco, sacramento, pointReyes, redding, medford, eugene, salem, portland, seattle, vancouver
total nodes in path :38
total distance : 5505.0

A*

explored nodes : miami, westPalmBeach, orlando, daytonaBeach, jacksonville, tampa, lakeCity, savannah, tallahassee, pensacola, augusta, albanyGA, macon, atlanta, charlotte, newOrleans, chattanooga, batonRouge, nashville, lafayette, beaumont, greensboro, houston, memphis, mexia, dallas, austin, littleRock, tulsa, raleigh, denver, sanAntonio, oklahomaCity, coloradoSprings, grandJunction, kansasCity, keyWest, elPaso, provo, norfolk, richmond, laredo, albuquerque, wichita, washington, santaFe, baltimore, tucson, phoenix, ftWorth, philadelphia, lincoln, omaha, yuma, pittsburgh, cleveland, stLouis, newYork, sanDiego, losAngeles, bakersfield, fresno, modesto, columbus, stockton, desMoines, sacramento, sanLuisObispo, dayton, salinas, sanJose, oakland, cincinnati, reno, sanFrancisco, lasVegas, buffalo, pointReyes, indianapolis, minneapolis, toronto, redding, medford, rochester, eugene, salem, portland, seattle
total explored nodes : 88
nodes in path : miami, westPalmBeach, orlando, tampa, lakeCity, tallahassee, pensacola, newOrleans, batonRouge, lafayette, beaumont, houston, austin, sanAntonio, elPaso, tucson, phoenix, yuma, sanDiego, losAngeles, bakersfield, fresno, modesto, stockton, sacramento, pointReyes, redding, medford, eugene, salem, portland, seattle, vancouver
total nodes in path :33
total distance : 4509.0

There are no such paths wherein the total number of explored nodes in Greedy is greater than A* algorithm.

5. There is no such path wherein the total cost to traverse from A to B in A* is greater than Dynamic algorithm.

Following is a path when the total number of explored nodes while traversing from A to B in Dynamic is greater than A* algorithm.

dynamic:

explored nodes : miami, westPalmBeach, orlando, daytonaBeach, tampa, jacksonville, lakeCity, savannah, tallahassee, augusta, pensacola, albanyGA, keyWest, charlotte, macon, atlanta, greensboro, raleigh, newOrleans, chattanooga, batonRouge, lafayette, nashville, norfolk, beaumont, richmond, houston, washington, memphis, baltimore,

mexia, austin, littleRock, philadelphia, dallas, sanAntonio, newYork, laredo, tulsa, pittsburgh, oklahomaCity, cleveland, kansasCity, ftWorth, columbus, elPaso, buffalo, dayton, wichita, rochester, cincinnati, toronto, stLouis, denver, indianapolis, albanyNY, coloradoSprings, albuquerque, mexico, tucson, lincoln, ottawa, boston, santaFe, omaha, phoenix, grandJunction, providence, montreal, desMoines, newHaven, saultSteMarie, yuma, stamford, provo, minneapolis, sanDiego, losAngeles, thunderBay, bakersfield, greenBay, sanLuisObispo, fresno, lasVegas, milwaukee, salinas, modesto, winnipeg, sanJose, stockton, chicago, oakland, sanFrancisco, sacramento, pointReyes, reno, midland, redding, toledo, saltLakeCity, medford, eugene, boise, salem, calgary, portland, seattle
total explored nodes : 107
nodes in path : miami, westPalmBeach, orlando, tampa, lakeCity, tallahassee, pensacola, newOrleans, batonRouge, lafayette, beaumont, houston, austin, sanAntonio, elPaso, tucson, phoenix, yuma, sanDiego, losAngeles, bakersfield, fresno, modesto, stockton, sacramento, pointReyes, redding, medford, eugene, salem, portland, seattle, vancouver
total nodes in path :33
total distance : 4509.0

A*

explored nodes : miami, westPalmBeach, orlando, daytonaBeach, jacksonville, tampa, lakeCity, savannah, tallahassee, pensacola, augusta, albanyGA, macon, atlanta, charlotte, newOrleans, chattanooga, batonRouge, nashville, lafayette, beaumont, greensboro, houston, memphis, mexia, dallas, austin, littleRock, tulsa, raleigh, denver, sanAntonio, oklahomaCity, coloradoSprings, grandJunction, kansasCity, keyWest, elPaso, provo, norfolk, richmond, laredo, albuquerque, wichita, washington, santaFe, baltimore, tucson, phoenix, ftWorth, philadelphia, lincoln, omaha, yuma, pittsburgh, cleveland, stLouis, newYork, sanDiego, losAngeles, bakersfield, fresno, modesto, columbus, stockton, desMoines, sacramento, sanLuisObispo, dayton, salinas, sanJose, oakland, cincinnati, reno, sanFrancisco, lasVegas, buffalo, pointReyes, indianapolis, minneapolis, toronto, redding, medford, rochester, eugene, salem, portland, seattle
total explored nodes : 88
nodes in path : miami, westPalmBeach, orlando, tampa, lakeCity, tallahassee, pensacola, newOrleans, batonRouge, lafayette, beaumont, houston, austin, sanAntonio, elPaso, tucson, phoenix, yuma, sanDiego, losAngeles, bakersfield, fresno, modesto, stockton, sacramento, pointReyes, redding, medford, eugene, salem, portland, seattle, vancouver
total nodes in path :33
total distance : 4509.0

6. Lets consider output of traversal from Vancouver to Miami:

A*

explored nodes : vancouver, seattle, portland, salem, boise, calgary, eugene, saltLakeCity, medford, winnipeg, redding, minneapolis, desMoines, thunderBay, greenBay, pointReyes, milwaukee, chicago, sacramento, stockton, modesto, reno, omaha, fresno, lincoln, bakersfield, lasVegas, sanFrancisco, oakland, saultSteMarie, sanJose, salinas, losAngeles, sanLuisObispo, sanDiego, yuma, wichita, phoenix, midland, toledo, tucson, toronto, elPaso, buffalo, kansasCity, sanAntonio, stLouis, rochester, cleveland, austin, houston, beaumont, lafayette, columbus, batonRouge, newOrleans, pittsburgh, tulsa, dayton, cincinnati, littleRock, albanyNY, laredo, pensacola, tallahassee, indianapolis, memphis, albuquerque, lakeCity, santaFe, oklahomaCity, tampa, ottawa, mexia, jacksonville, daytonaBeach, orlando, nashville, boston, albanyGA, westPalmBeach
total explored nodes : 81
nodes in path : vancouver, seattle, portland, salem, eugene, medford, redding, pointReyes, sacramento, stockton, modesto, fresno, bakersfield, losAngeles, sanDiego, yuma, phoenix, tucson, elPaso, sanAntonio, austin, houston, beaumont, lafayette, batonRouge, newOrleans, pensacola, tallahassee, lakeCity, tampa, orlando, westPalmBeach, miami
total nodes in path :33
total distance : 4509.0

Greedy

explored nodes : vancouver, calgary, winnipeg, minneapolis, desMoines, greenBay, milwaukee, chicago, midland, toledo, omaha, lincoln, wichita, kansasCity, stLouis, indianapolis, cincinnati, dayton, columbus, cleveland, pittsburgh, philadelphia, baltimore, washington, richmond, norfolk, raleigh, greensboro, charlotte, augusta, savannah, jacksonville, daytonaBeach, orlando, westPalmBeach
total explored nodes : 35
nodes in path : vancouver, calgary, winnipeg, minneapolis, desMoines, omaha, lincoln, wichita, kansasCity, stLouis, indianapolis, cincinnati, dayton, columbus, cleveland, pittsburgh, philadelphia, baltimore, washington, richmond, norfolk, raleigh, greensboro, charlotte, augusta, savannah, jacksonville, daytonaBeach, orlando, westPalmBeach, miami
total nodes in path :31
total distance : 5657.0

Dynamic

explored nodes : vancouver, seattle, portland, salem, eugene, medford, calgary, redding, boise, pointReyes, sacramento, saltLakeCity, stockton, modesto, sanFrancisco, oakland, reno, sanJose, salinas, fresno, bakersfield, sanLuisObispo, winnipeg, losAngeles, lasVegas, sanDiego, yuma, thunderBay, minneapolis, phoenix, tucson, desMoines, greenBay, omaha, saultSteMarie, milwaukee, lincoln, elPaso, chicago, wichita, albuquerque, midland, santaFe, toronto, toledo, kansasCity, buffalo, rochester, sanAntonio, coloradoSprings, ottawa, austin, cleveland, tulsa, stLouis, albanyNY, denver, laredo, montreal, oklahomaCity, columbus, pittsburgh, houston, boston, dayton, beaumont, providence, indianapolis, cincinnati, grandJunction,

littleRock, ftWorth, mexia, newHaven, lafayette, batonRouge, dallas, memphis, stamford, philadelphia, newOrleans, provo, newYork, baltimore, washington, nashville, richmond, pensacola, chattanooga, mexico, norfolk, tallahassee, atlanta, macon, lakeCity, albanyGA, raleigh, greensboro, jacksonville, tampa, charlotte, daytonaBeach, savannah, orlando, augusta, westPalmBeach
total explored nodes : 106
nodes in path : vancouver, seattle, portland, salem, eugene, medford, redding, pointReyes, sacramento, stockton, modesto, fresno, bakersfield, losAngeles, sanDiego, yuma, phoenix, tucson, elPaso, sanAntonio, austin, houston, beaumont, lafayette, batonRouge, newOrleans, pensacola, tallahassee, lakeCity, tampa, orlando, westPalmBeach, miami
total nodes in path :33
total distance : 4509.0

In all cases dynamic algorithm and A* gives least cost to travel from source to destination. Since dynamic algorithm explored all the possible path to traverse from source to destination, the number of explored nodes will be greater. However in case of A*, it adds heuristic aspect while selecting frontier, this helps in minimizing number of explored nodes. In case of Greedy, it calculates least path cost required to reach destination at every node. That is, it finds locally optimal solution in the hope to find globally optimum result. However, this is not always the case. Therefore in most cases cost to traverse from source to destination is always greater in Greedy as compared to that of dynamic.

References:

[1] Building Watson: An Overview of the DeepQA Project. (pg. 68)