Tutorial - 1

1. Explore the spectrum from static to AI based techniques for the problem given below The program should increase in complexity, generalization, clarity of knowledge and extensibility of their approach.

A. Tic-Tac-Toe

B. Question Answering (Reading an English text and answer the question)

C. Language Translation (Translating English sentence into hindi/any other language).

→ A. Tic-Tac-Toe.

– Program-1

• Board data structures:

Board:

A nine-element vector representing the board, where elements of vector correspond to board position as follows:

```
1 2 3 ⟶  1 2 3 4 5 6 7 8 9
4 5 6
7 8 9
```

9-element vector.

Each element will contain,
- 0 for blank
- 1 for 'X' player move
- 2 for 'O' player move.

• Movetable:

It is a vector of $3^9 = 19,683$ elements, each element of which is a nine-element vector.

Index	Current Board	New Board
0	000000000	000010000
1	000000001	002000001
2	000000002	000100002
3	000000010	002000010
⋮		
19683		

• Algorithm:

To make a move, do the following:

1. View vector board as a ternary number. Convert it into a decimal number.
2. Use number computed in above step as an index into movetable and access the vector stored there
3. The vector selected in step 2 represents the way the board will look after the move that should be made. So set board equal to that vector.

• Advantages:

- Very efficient in terms of time. And, in theory it could play an optimal tic-tac-toe game.

• Disadvantages:

- It takes a lot of space to store the Move-table.
- Requires a lot of work to specify all entries in move table
- Difficult to extend for higher dimensions
- Highly error prone as the data is voluminous
- Poor extensibility.

— Program-2:

• Data Structure:

Board: A

A nine-element vector representing the board, as where elements of vector correspond to board position as follows:

8	3	4
1	5	9
6	7	2

These board positions have the sum of all rows, columns and diagonals is 15.

PAGE NO.
DATE / /

• Algorithm:

1. In addition to marking the board as moves are made, we keep a list, for each player, of the square in which he/she has played.
2. To check the possibility of win for one player, we consider each pair of square owned by that player and compute the difference between 15 and the sum of two square.
3. If difference is negative or it is greater than 9, so ignore that pair.
4. Else if the square representing the difference is blank, a move there will produce a win.

• Advantages:

- It is more efficient in terms of space.
- It is also lot easier to understand program's strategy or to change strategy if desired.

• Disadvantages:

- This program is not quite as effiecient in terms of time as the first one.
- Difficult to generalize program's knowledge for three-dimensional tic-tac-toe.

— Program-3:

• Data Structure:

Board:

A structure containing a nine-element vector representing the board, a list of board positions that could result from the next move, and a number representing an estimate of how likely the board position is to lead to an ultimate win for the player to move.

• Algorithm:
To decide on the next move, look ahead at the board positions that result from each possible move. Decide which position is best, make the move that leads to the position, & assign the rating of that best move to current position. To decide best position follow the following steps:
1. See if it is a win. If so, call it the best by giving it the highest possible rating.
2. Otherwise, consider all the moves the opponent could make next. See which of them is worst for us. Assume the opponent will make that move. Whatever rating that move has, assign it to the node we are considering.
3. The best node is then the one with the highest rating.

• ~~Advantages~~
• Advantages:
- This approach is extensible to handle 3 dimensional tic-tac-toe and it could be extended to handle games which are more complicated than tic-tac-toe

• Disadvantages:
- This program will require more time than two others as it has to search a tree representing all possible move sequences before making each move.

B. Question Answering
- Program-1:
• Data Structures:
A set of templates that match common questions and produce patterns used to match against inputs. Templates and patterns are used so that a template that matches a given question is associated with the corresponding

pattern to find the answer in the input text.
Eg, The template who did x y generates x y z if a match occurs and z is the answer to the question. The given text and the question are both stored as strings.

• Algorithm:
1. Compare the template against the questions and store all successfull matches to produce a set a text patterns
2. Pass these text patterns through a substitution process to change the person or voice and produce an expanded set of text patterns.
3. Apply each of these patterns to text; collect all answer and print each answer.

• Comments:
- Very primitive approach and worse than that used in the game.
- This type of technique was actually used in ELIZA.

— Program 2:
• Data Structures:
A structure called English consists of a dictionary, grammar and some semantics about the vocabulary. This data structures provides the knowledge to convert english text into storable internal form and also to convert the response back into English. The structured representation of text is a processed form and defines the content of the input text by making explicit all references.
There are three types of ~~each~~ such knowledge representation system: production rules of the form 'if x then y', slot and filler system and statements in mathematical logic.

PAGE NO.
DATE / /

• Algorithm:
1. Convert the question to a structured form using English know how, then use a marker to indicate the substring of the structure, that should be returned as an answer. If a slot and filler system is used a special marker can be placed in more than one slot.
2. The answer appears by matching this structured from against structured text.
3. The structured form is matched against the text and the requested segment of the question are returned.

• Comments:
- This approach is more meaningful than previous one and is more effective.
- The problem of handling pronouns are difficult.
- The extra power given must be paid for by additional search time in knowledge bases.

— Program-3:
• Data Structure:
World model is a structured representation of background world knowledge which contains knowledge about objects, actions and situations that are described in the input text. This structure is used to create integrated text from input text. This information is known as script.

• Algorithm:
1. Convert the question to a structured form using both the knowledge and ~~no~~ world model by generating even more possible structures.
2. The structured form is matched against the text and the requested segments of questions are returned.

PAGE NO.
DATE / /

• Comments:
- It has more knowledge than previous program hence it is more powerful.
- The major omission is that of a general ~~reasoning~~ reasoning mechanism known as inference to be used when the required answer is not explicitly given in input text.

C. Language Translation:
- Program 1:
• Data Structure:
A set of pre-defined rules, which are represented in formal grammar. The rules are stored in a database or file. The rules consist of words in alphabetical order (lexicon of words) and grammar of language with its translation in other language.

• Algorithm:
1. Tokenize the input text into words and phrases using natural language processing techniques
2. For each token match it with the lexicon of word in predefined rules.
3. For matched words, apply the grammer rules to form the target language phrase
4. Concatenate all the formed phrases to generate the final translation.

• Comments:
- This approach is less complex as it is based on pre-defined rules and simple pattern matching thus easy to interpret
- It does not consider context or meaning of text.
- This approach is specific to the language whose rule is defined in file & thus cannot be applied to other language.

PAGE NO.
DATE / /

— Program 2:

• Data Structure:

The data structure used in this approach is deep neural network, trained on a large corpus of parallel text. The corpus is used to train the model. The neural network is represented as a series of interconnected layers.

• Algorithm:

1. Tokenize the input text into words and phrases
2. Use deep neural network to generate target text.
3. Repeat the process for all tokens.

• Comments:

- This approach is more efficient than previous one but it is highly complex, as this approach is based on deep learning neural network.
- Easy to generalize this approach for other languages and can generate translations with high accuracy & fluency.
- Clarity of knowledge is low as models are highly complex and difficult to interpret.