Dine-Out Problem

Prashant Kagwad

Problem Description

A user can have a hard time in the selection of dine-out places like diner/restaurant/food-joints given certain preferences. The preferences can be food quality [scale of 1 - 5], cuisine [Italian/Chinese/Indian], food content [meat/allergic items], price [low, high, affordable], atmosphere [kid-friendly/bachelors/crowded], convenience factor [distance], etc.

The objective of the problem is to find the best fit dine-out place for the user based on the preference that he/she may provide. To achieve this, a logical agent will accept user's location & preferences for dine-out and in return provide a travel route from user's location to dine-out location that satisfies the preferences.

Proposed Solution

Informed search algorithm:

To perform the informed search, A* search algorithm is used to search the closest best fit diner to the user's location based on the preferences given. [Finding shortest distance between 2 locations on a graph/map.]

Two (2) heuristics (one admissible and another inadmissible):

1. Distance between two locations:

Straight line distance for current location to selected dine-out location (admissible heuristics i.e. it will always underestimate the actual distance between the two locations)

When expressed in terms of an evaluation function this would be expressed as:

$$f^*(n) = g^*(n) + h^*(n)$$

estimates the minimum distance from source to nth node and from nth node to destination, by expanding the node that minimizes the remaining distance to destination

2. Time taken to travel between two locations:

Time taken to reach the destination (inadmissible i.e. as the time take from one location to other location may change due to the environmental conditions and may overestimate the actual time to reach the destination)

When expressed in terms of an evaluation function this would be expressed as:

$$f(n) = h(n)$$

estimates the time from nth node to destination, by expanding the node that appears to be closest to destination

Implementation Details

Example:

To run the program we will be using the following options

DineOut – Project Title

Option 1 [`Item Search`]*: True – Item base search, False – Restaurant based search

Option 2 [`Category`]*: Veg – Vegetarian, Non-Veg – Non-Vegetarian /Meat

Option 3 [`Heuristics`]*: Dist – Distance, Time – Time based

Option 4 [`Preferences`]: User preferences like - Pizza||Music||..

Example 1: Using admissible heuristics – Distance

Input: DineOut True Veg Dist

Output:

```
itemBasedSearch : true
Category : Veg
Heuristics : Dist
User Preferences : None


Matched Items :
================================================================================
1. Caffe Moca
       Served At : Starbucks
       Contents  : [Coffee||Milk]

2. Cheese Burger
       Served At : Chillies
       Contents  : [Bread||Cheese||Vegetables]

3. Italian Pasta
       Served At : CiCi's Pizza
       Contents  : [Pasta||Cheese||Vegetables||Tomato Sauce]
```

```
4. Veggi Salad
       Served At : Subway
       Contents  : [Vegetables||Dressing]

================================================================================


Matched Resturants :
================================================================================
1. Starbucks
       Cuisine  : [Fastfood||Coffee||Salad||Juice||Tea]
       Ambience : [Bachelors||Crowded||Wi-Fi||Music]

2. Chillies
       Cuisine  : [Dine-In||Mexican||Non-veg]
       Ambience : [Dine-In||Crowded||Music||Kid-Friendly]

3. CiCi's Pizza
       Cuisine  : [Fastfood||Pizza||Salad||Pasta||Non-Veg||Veg]
       Ambience : [Dine-In||Family]

4. Subway
       Cuisine  : [Fastfood||Sandwich||Salad||Dessert||Pizza]
       Ambience : [Music||Dine-In||Family]

================================================================================


Path to Resturants :
================================================================================
1. Path from Home to Starbucks :
       [Home, McCallum Blvd, W Renner Rd, W Campbell Rd, Coit Rd, Starbucks]

2. Path from Home to Chillies :
       [Home, McCallum Blvd, W Renner Rd, W Campbell Rd, Chillies]

3. Path from Home to Cici's Pizza :
       [Home, McCallum Blvd, Waterview Pkwy, Cici's Pizza]

4. Path from Home to Subway :
       [Home, McCallum Blvd, W Renner Rd, W Campbell Rd, Coit Rd, Subway]

================================================================================
```

Example 2: Using inadmissible heuristics – Time

Input: DineOut True Veg Time

Output:

```
itemBasedSearch : true
Category : Veg
Heuristics : Time
User Preferences : None


Matched Items :
================================================================================
1. Caffe Moca
       Served At : Starbucks
       Contents  : [Coffee||Milk]

2. Cheese Burger
       Served At : Chillies
       Contents  : [Bread||Cheese||Vegetables]

3. Italian Pasta
       Served At : CiCi's Pizza
       Contents  : [Pasta||Cheese||Vegetables||Tomato Sauce]

4. Veggi Salad
       Served At : Subway
       Contents  : [Vegetables||Dressing]

================================================================================


Matched Resturants :
================================================================================
1. Starbucks
       Cuisine  : [Fastfood||Coffee||Salad||Juice||Tea]
       Ambience : [Bachelors||Crowded||Wi-Fi||Music]

2. Chillies
       Cuisine  : [Dine-In||Mexican||Non-veg]
       Ambience : [Dine-In||Crowded||Music||Kid-Friendly]

3. CiCi's Pizza
       Cuisine  : [Fastfood||Pizza||Salad||Pasta||Non-Veg||Veg]
       Ambience : [Dine-In||Family]

4. Subway
       Cuisine  : [Fastfood||Sandwich||Salad||Dessert||Pizza]
       Ambience : [Music||Dine-In||Family]

================================================================================
```

```
Path to Resturants :
================================================================================
1. Path from Home to Starbucks :
      [Home, McCallum Blvd, Waterview Pkwy, Cici's Pizza, Custer Pkwy, Parker Rd,
Starbucks]

2. Path from Home to Chillies :
      [Home, Frankford Rd, W Campbell Rd, Chillies]

3. Path from Home to Cici's Pizza :
      [Home, McCallum Blvd, Waterview Pkwy, Cici's Pizza]

4. Path from Home to Subway :
      [Home, Frankford Rd, W Campbell Rd, Coit Rd, Subway]

================================================================================
```

Knowledge-base (in OWL):

1) “Veggi Salad” is_type_of “Salad”

2) “Salad” is_served_at “Subway”

3) “Dessert” is_served_at “Subway”

4) ”Subway” is_at “Location_Subway”

5) “Location_Subway” location_details Location_Object

6) So on……

Inferred Facts:

1) “Subway” is_located_at Location_Object

2) “Veggi Salad” is_served_in “Subway”

3) So on…..

Programming tools (including third party software tools to be used):

Some of the third party programing tools/frameworks that were used:

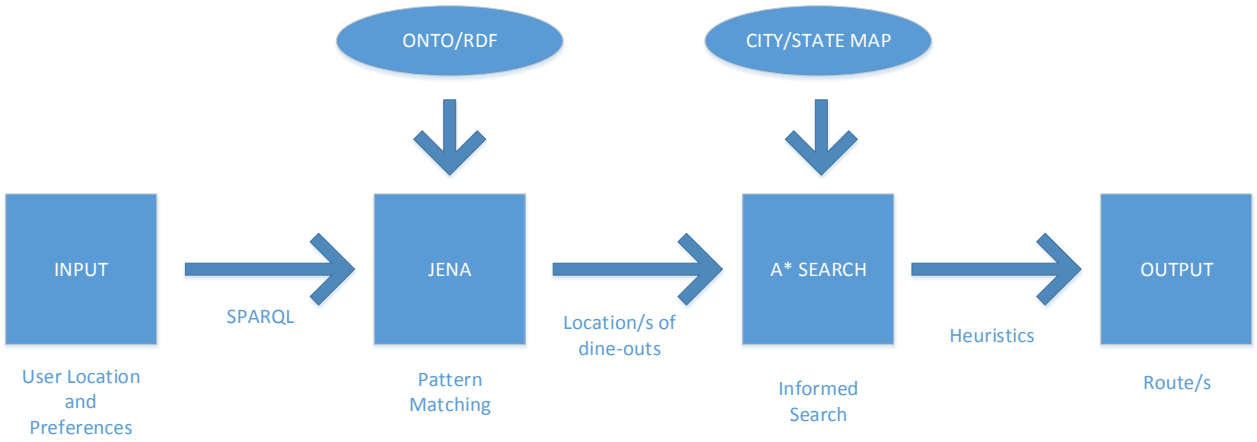
Protégé – to construct the domain model and necessary knowledge-base for the application (ontologies - XML)

Jena – Jena library for java. This was used to consume and query the data in knowledge-base. The library contains an RDF/XML parser/serializer that conforms to the RDF/XML Syntax Specification.

Fuseki-Server – to test the SPARQL queries used to extract datainformation from OWL/RDF that are stored online (locations mentioned in the architecture diagram):

Java – to implementation of the process/agent, search algorithm and stub code to consume and query the data/information stored at OWL/RDF.

Architectural Diagram:



The input to the program were the user preferences [location was implicitly mentioned]. The program stared by first consuming and querying the ontologies created to store information about user, restaurants and locations of every entity. The ontologies were made using protégé and stored online at the following location:

User Data: http://www.utdallas.edu/~pdk130030/owl/user.owl

Item/Restaurant Data: http://www.utdallas.edu/~pdk130030/owl/resturant.owl

Location Data: http://www.utdallas.edu/~pdk130030/owl/location.owl

Once the data is queried on restaurant ontology, the program computes the shortest distances between user location and destination using the desired heuristics mentioned as part of the input. Finally based on user preferences, the program outputs list of items, restaurants that match the preference and a list of paths that can be taken in order to reach all the restaurants.

Results:

The program provides with a list of food items served at different food joints and path to all the food joints (restaurants) from user's location in the map.

Problems:

There were only a few minor problems (issues) that were faced during the construction of this project. Most of them were related to configuration related i.e. making different components to work with each other.

Also most of the time was spent creating the ontology using Protégé in a way that was suitable to query using SPARQL. And addition of data to cover different scenarios was also time consuming. Understanding the construction of the ontology i.e. KB, gives a developer the insights that in the field of AI it is the most important feature alongside implementation of searching algorithm.

Potential Improvements:

The searching and inference implantation works amazing well, but with the addition of different parameters will require to implement better inferences which will certainly improve the overall performance of the program.

Addition of more data will provide the program with better capabilities to parse data better and find specific results more easily and accurately with respect to user preferences.