Where Do We Meet?

Coursera Capstone
Project Report

February 2019

Introduction

We all have families, friends and business contacts around the country. Last decade we started to meet online, but meeting in a common place is always preferred, and sometimes is essential. In that case, the procedure depends on a few things;

- How far are the groups (3 km, 15 miles etc.),
- How is the transportation (by car, by public transportation etc.),
- Do groups have special needs/preferences in a meeting place (children, restaurant, disabled people etc.),
- Is there an appropriate place in between, within the transportation?

Generally, after a quick assessment, the meeting place is chosen between known places. If the middle point is an unknown place or if a change on the meeting place is intended, then some search on the internet will help to determine the place.

In this frame, social interaction would be the main need to use this tool. Also, it would be used with business purposes such as finding a warehouse, store or maintenance place for stores in various places, or for marketing purposes. So any individual user may be in a situation to use this tool. Specifically, we intend to specify the problem to refugee (asylum seeker) groups in Netherlands.

Problem Description

The problem that we focus is about shortening the search time to find the appropriate city and places to meet.

To be more precise, if two or more groups around the country intend to find an appropriate meeting point (children playground, hotel, restaurant, sports bar etc.) in a city in-between current group locations, a tool which takes the current locations and gives the suggestions would help a lot.

The audience in this project is AZC guests (refugees and asylum seekers). Since these people are new in the country, with no prior experience and lack of language, they generally struggle to find an appropriate place. Most of the guests don’t even know their

locations in the country (just know the name of AZC) and can not estimate a middle point to meet.

In the described frame, I would like to find a solution to the problem described above. So in Netherlands , there are community centers for refugees, called AZC. These centers are located all over Netherlands and people may move from one to another. When someone wants to meet with a friend from another AZC, since the refugees would not have a car, she generally tries to **find an in-between city**, an appropriate **place to meet** in the city (spend time) and travel there by **public transportation**.

Data

In this project, the data from COA website (official AZC institution) was used to gather AZC locations. Also, foursquare data was used to spot an appropriate place around intended meeting point.

First, the webpage was downloaded, and scraped to reach the exact latitude and longitudes of AZC's (reception centers). Then two random AZC locations was picked to simulate two groups. Then, with a few equations, the middle point was determined with a radius large enough to a public transportation. Lastly, after finding the approximate meeting point, I would use the foursquare data to list the categories and alternative places to meet.

Gather and Scrape the Location Data

The full webpage was written in Java and it's complicated to reach the location info with an automated process. I think it was easier to scrape the information with a text editor using regex, so we used a text editor for faster process. An output file was generated containing the lat-long coordinates of the 56 AZC's.

Convert the Data to Dataframe

The coordinates for current locations in the file was converted to a dataframe.

Methodology

First we checked the data to see how it looks on the map.

Plot the Location Data On a Map

Fig. 1 - All AZC locations, Netherlands

From the data consisting is coordinates of 56 AZC's, two random places were picked (actually not random, but that information is out of scope now) to simulate an example.

Latitude	Longitude	Name
51.2908	5.62967	Budel-Cranendonck
52.1460	4.38730	Wassenaar-Duinrell

Find a Middle Point

After choosing two random group points to simulate current places, we find the middle point.

51.7184	5.00848	Middle Point

We check it again on the map. We now use a detailed map (openstreetmap) to see the main highways and train rails to understand the public transportation assets to middle point.

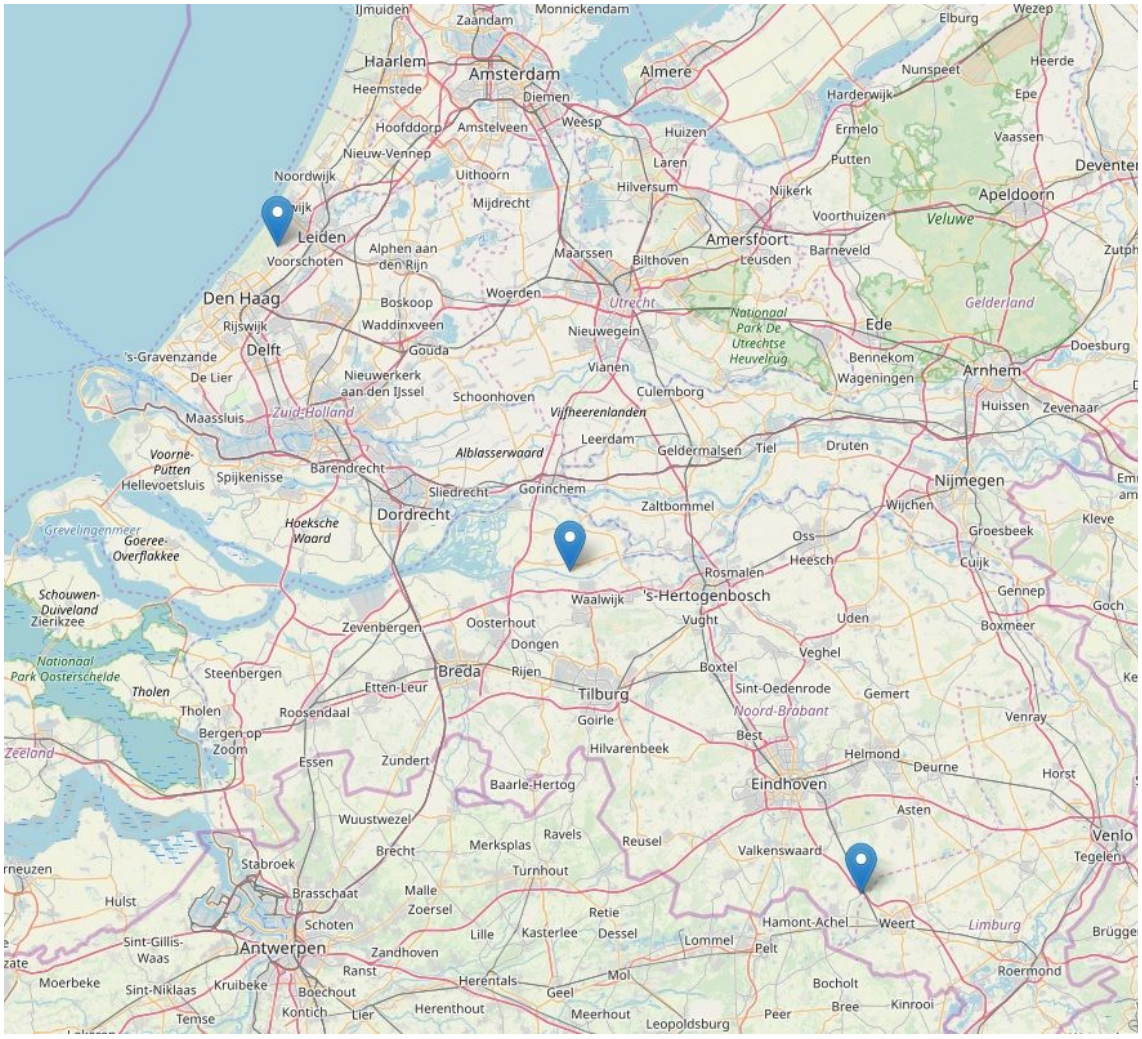

Fig. 2 - Location of each group and the middle point

We can see on the map that the midpoint is not in the center of a city, so we need to estimate a close (in 10 km) city center with public transportation (train especially).

Foursquare Data to Overcome Transportation

We used the term 'midpoint' as the point that the temporary place to scan around, for a public transportation stop. After finding an appropriate public transportation point, we moved the midpoint to the station, and started to scan for an appropriate meeting point. We specified the range up to 1 kms. Below are the responses from foursquare server when we looked for a *'bus'* or *'train'* category near the midpoint.

0 : Train Station - Station De Oost
1 : Taxi - Station De Oost
2 : Bus Line - Station De Oost
3 : Bus Stop - Station De Oost
4 : Bus Station - Station De Oost

The first result is an appropriate train station to put it on the center. So we make the index 0 our new midpoint.

Train Station (Station De Oost) - lat: 51.61~ , long: 5.05~

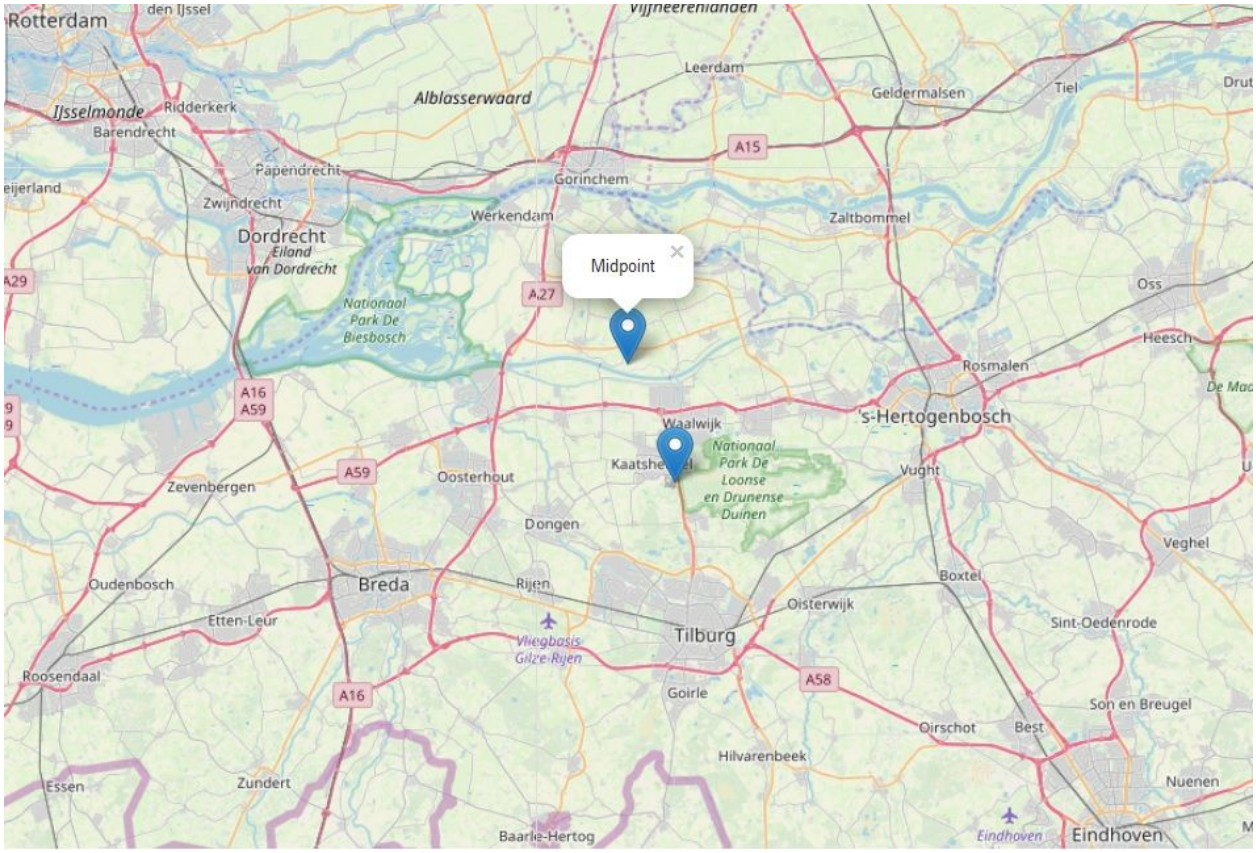

Fig. 3 - The coordinates of the midpoint and the closest train station (Station De Oost)

Foursquare Data for Appropriate Places

When we asked for 50 nearby popular places in vicinity of the station, foursquare server sends us back 49 places that are from diverse categories. This is much better because now we have the flexibility to choose.

These places are:
0 : Theme Park Ride / Attraction - Joris en de Draak
1 : Theme Park Ride / Attraction - De Vliegende Hollander
2 : Theme Park Ride / Attraction - Baron 1898
3 : Theme Park Ride / Attraction - Python
4 : Theme Park Ride / Attraction - Ruigrijk
5 : Theme Park Ride / Attraction - D'Oude Tuffer
6 : Theme Park Ride / Attraction - Halve Maen
7 : Theme Park Ride / Attraction - Piraña
8 : Theme Park - Efteling
9 : Theme Park Ride / Attraction - Symbolica: Paleis der Fantasie
10 : Theme Park - De Blauwe Reiger
11 : Theme Park Ride / Attraction - Pagode
12 : Train Station - Station De Oost
13 : Theme Park Ride / Attraction - Vogel Rok
14 : Creperie - Polles Keuken
15 : Theme Park Ride / Attraction - Bob
16 : Theme Park Ride / Attraction - Gondoletta
17 : Theme Park Ride / Attraction - Sprookjesbos
18 : Theme Park Ride / Attraction - Diorama
19 : Theme Park Ride / Attraction - Aquanura
20 : Theme Park Ride / Attraction - Fata Morgana
21 : Hotel - Efteling Hotel
22 : Plaza - Carnaval Festivalplein
23 : Theme Park Ride / Attraction - Spookslot
24 : Theme Park Ride / Attraction - Carnaval Festival
25 : History Museum - Efteling Museum
26 : Sandwich Place - Carrousel Paleis
27 : Food Truck - De Eigenheymer
28 : Plaza - Anton Pieckplein
29 : Theme Park Ride / Attraction - Roodkapje
30 : Café - Wachtruimte 1e Klas
31 : Theme Park Ride / Attraction - Marerijk
32 : Theme Park Ride / Attraction - De Indische Waterlelies
33 : Salad Place - De Verse Oogst
34 : Theme Park Ride / Attraction - Droomvlucht

35 : Candy Store - De Verleiding
36 : Theme Park Ride / Attraction - Draak Lichtgeraakt
37 : Bar - Bar de Gelagkamer
38 : Gift Shop - Jokies Wereld
39 : Snack Place - De Hongerige Machinist
40 : Theme Park Ride / Attraction - Villa Volta
41 : Snack Place - De Witte Walvis
42 : Theme Park Ride / Attraction - Doornroosje
43 : Playground - IJspaleis
44 : Pizza Place - 't Melkhuysje
45 : Theme Park Ride / Attraction - Sprookjesboom
46 : Building - Villa Pardoes
47 : Resort - Vakantiepark Efteling Loonsche Land
48 : Theater - Raveleijn
49 : Theme Park Ride / Attraction - De Trollenkoning

Find Appropriate Places

We choose places for families with children, in a sequence that is adaptable to real life (entertain, food, snack loop).

So No.12, No.30, No.43, No.44, No.25, No.35, No.27, No.12 may be the route for this trip. Which means:

Train Station - Café - Playground - Pizza Place - History Museum -Food Truck - Train Station

Find Walking Distances

Lastly, we checked if walking distances are in humanity limits. The walk distances from one point to another in meters is as follows:

Train Station - Café - Playground - Pizza Place - History Museum - Food Truck - Train Station
(365 m.) (518 m.) (239 m.) (368 m.) (169 m.) (485 m.)

We need t point out that the coordinates may vary in this scale, and these are bird-eye distances. Even for that the walking distances looks ok.

Results

We aimed for a tool to answer ‘Where Do We Meet?’ and we answered the question in detail. The example was for two groups in different places of Netherlands, and they ended up in a mid-place, which is near a train station approximately has the same distance to each group. We found a few places and a route to enjoy these places in a weekend for these two groups.

The results are so convincing and the data was gathered from real world, so it is likely that this route will be used by two groups :)

Discussion

As was described in the first chapter, this example is just one side of using this tool. This is a social need, to find a place to go in a short time. So it can be used by one individual with a child, the ‘midpoint’ can be the new school and the filter word can be ‘playground’ instead of ‘station’.

Another use case can be three people, that agree on the meeting city but cannot agree on the exact place can end up this discussion with agreeing on the random ‘sports bar’ recommandation of the program.

Also, on the business side, the audience expects to see a ‘trending place’ or ‘in sequence’ place which is the place visited after a certain point, in the vicinity of their supermarket chain.

Conclusion

To sum up, *the problem* about shortening the search time to find the appropriate city and places to meet for AZC guests in Netherland has a relative solution with this tool. It indeed shortens the search time.

Further improvements on the user interface (or an integration to COA app) would generalize the usability of the tool.