Battle of Neighborhoods:Explore the Similarity of Given Metropolitan Cities

Author:x.l.dang@Beijing,China. Feedback Email:xldang2020@gmail.com (mailto:xldang2020@gmail.com)

Table of contents

Chapter 1. Battle understanding

In this assignment, I explored New York City ,Toronto and other cities with their respective segmented and clustered neighborhoods. One interesting idea I explored is comparing the neighborhoods of the two cities and determine how similar or dissimilar they are. For example, is New York City more like Toronto or Paris or some other multicultural city?

Chapter 2. Data Requirements

For given metropolitan cities,I will explore the similarity of them from the perspective of venue distribution.The venues of a city are like human capillaries. Each venue is a small feature of the city. Together, these features can reflect a certain aspect of the city.

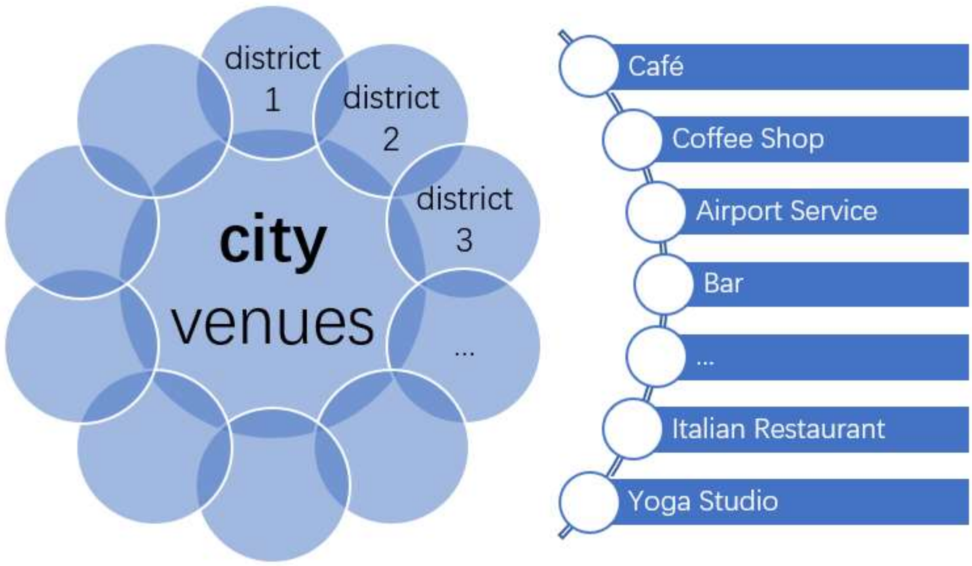

Fig.1 **City capillaries**

As shown in **Fig.1**, for a city, I need to collect its neighborhood(or district) data and the Venue data of each neighborhood.

Chapter 3. Data collection

In order to obtain the target data, I executed the following path:

- Crawling wiki pages or other pages containing city district information
- Perform data cleaning on crawled webpages
- Obtain (Latitude,Longitude) infromation for each neighborhood/district using python toolkit <geopy.geocoders.Nominatim>
- Extracting venue information for each (Latitude,Longitude) using Python API < FourSquare >
- Each city corresponds to a pandas.DataFrame that stores target data.

3.1 import python dependency packages

I need to import:

- requests library to handle requests,
- pandas library for data analsysis,
- numpy library to handle data in a vectorized manner,
- geopy.geocoders module to convert an address into latitude and longitude values,
- and plotting librarys include seaborn and matplotlib.pyplot.

3.2 Geography information of chosen metropolitan cities

Without any prejudice, I chose Shanghai, London, New York City, Frankfurt, and Toronto from five countries respectively: China, the United Kingdom, the United States, Germany, and Canada.

- Shanghai,China

- London, the United Kingdom
- New York City, the United States
- Frankfurt, Germany
- Toronto,Canada

3.2.1 A tiny case

Obtain the (Latitude,Longitude) infromation of location<E1,Head district,London>

```
postal_code,district='E1','Head district'
address ='{},{},London'.format(postal_code,district)
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
```

Obtain the venue infromation of location (latitude, longitude).

```
CLIENT_ID = 'QYHRIW4L1Y5WF2MAPABDG5WNS4CVTPBPM5GJ35W5R02SLMB2' # your Foursquare ID
CLIENT_SECRET = 'SAKZYG1K2PN1Z0V1VXFZZYHQF33DAZBNOTALAHYY0UWPYSL0' # your Foursquare
Secret
VERSION = '20180604'
radius=500
LIMIT = 30
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius=
{}&limit={}'.format (https://api.foursquare.com/v2/venues/explore?
&client_id=%7B%7D&client_secret=%7B%7D&v=%7B%7D&ll=%7B%7D,%7B%7D&radius=%7B%7D&limit=%7E
(
CLIENT_ID,
CLIENT_SECRET,
VERSION,
latitude,
longitude,
radius,
LIMIT)
results = requests.get(url).json()["response"]['groups'][0]['items']
results[0]['venue'] # take a look
```

3.2.2 Help functions

Generate location by Postal Code: *gen_location_by_post(post_lis,district_lis,city='')*

Generate location by district name: *gen_location_by_district(district_lis,city='')*

Get Nearby Venues for a certain location(latitudes, longitudes): *getNearbyVenues(names, latitudes, longitudes,radius,LIMIT)*

3.2.3 Preparing London geography dataset

https://en.wikipedia.org/wiki/London_postal_district (https://en.wikipedia.org/wiki/London_postal_district)

Store the target data in a .csv file by crawling the above URL and read it by pd.read_csv

Such as London_POST=pd.read_csv('LONDON_POST.csv')

Use gen_location_by_post:

Location_with_post=gen_location_by_post(London_POST['Postal Code'].values,London_POST['District'].values,'London')

Merge the Location_with_post and the London_POST like this:

geo_merged_London=pd.merge(London_POST,Location_with_post,on='Postal Code')

**Then,store data locally like this: *

geo_merged_London.to_csv('geo_merged_London.csv',index=None)

3.2.4~3.2.7 follow with the same method.

3.2.4 Preparing New York geography dataset

https://en.wikipedia.org/wiki/Neighborhoods_in_New_York_City (https://en.wikipedia.org/wiki/Neighborhoods_in_New_York_City)

Store the target data in a .html file by crawling the above URL and read it by pd.read_html

3.2.5 Preparing Frankfurt geography dataset

https://en.wikipedia.org/wiki/List_of_Ortsbezirke_of_Frankfurt_am_Main (https://en.wikipedia.org/wiki/List_of_Ortsbezirke_of_Frankfurt_am_Main)

Store the target data in a .html file by crawling the above URL

3.2.6 Preparing Shanghai geography dataset

https://baike.baidu.com/item/%E4%B8%8A%E6%B5%B7%E8%A1%8C%E6%94%BF%E5%8C%BA%E5%88%9
fr=aladdin
(https://baike.baidu.com/item/%E4%B8%8A%E6%B5%B7%E8%A1%8C%E6%94%BF%E5%8C%BA%E5%88%
fr=aladdin)

Store the target data in a .html file by crawling the above URL

3.2.7 Preparing Toronto geography dataset

https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M
(https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M)

Since Toronto's data has been obtained in the coursework, it can be used directly here.It store in local<geo_merged_Toronto.csv>

3.3 Venue information of chosen metropolitan cities

3.3.1 Read prepared geo. dataset

Real local prepared dataset like this:
geo_merged_London=pd.read_csv('geo_merged_London.csv')
geo_merged_Torondo=pd.read_csv('geo_merged_Toronto.csv')
geo_merged_NewYork=pd.read_csv('geo_merged_NewYork.csv')
geo_merged_Frankfurt=pd.read_csv('geo_merged_Frankfurt.csv')
geo_merged_SH=pd.read_csv('ShangHai.csv')

3.3.2 Generate district nearby Venues

I use python API to do this:
CLIENT_ID = 'QYHRIW4L1Y5WF2MAPABDG5WNS4CVTPBPM5GJ35W5R02SLMB2' # your Foursquare ID
CLIENT_SECRET = 'SAKZYG1K2PN1Z0V1VXFZZYHQF33DAZBNOTALAHYY0UWPYSL0' # your Foursquare Secret
VERSION = '20180604'
and,the usage of the *getNearbyVenues* is follows this pattern:

toronto_venues = getNearbyVenues(names=geo_merged_Torondo['District'],
latitudes=geo_merged_Torondo['Latitude'],
longitudes=geo_merged_Torondo['Longitude'],
radius=500,
LIMIT=100
)

3.3.3 Venue dataset summary

Take a look at London_venues, it's shown that there 3429 **Venues are collected with total** 282 **categories.**

Chapter 4. Methodology

In this section, I will calculate the frequency of each venue category in each city. Then the similarity distance of each city can be obtained.

4.1 Load dataset

Read local venue dataset like this:
Toronto_venues=pd.read_csv('toronto_venues.csv')
SH_venues=pd.read_csv('SH_venues.csv')

NewYork_venues=pd.read_csv('NewYork_venues.csv')
London_venues=pd.read_csv('London_venues.csv')
Frankfurt_venues=pd.read_csv('Frankfurt_venues.csv')

I get these dataset with shape of :

- Toronto_venues.shape is (2115, 7)
- SH_venues.shape is (1128, 7)
- NewYork_venues.shape is (4665, 7)
- London_venues.shape is (3429, 7)
- Frankfurt_venues.shape is (1768, 7)

4.2 Modeling

Calculate the frequency of each venue category in each city

- Gather all Categories among these five cities and excludes all repeated categories.

- Calculate frequency by gen_Freqency function :*gen_Freqency(cityName,allCats,dFrame)*

- Merge values by pd.concat as follows,
  all_Freq=pd.concat([Shanghai_Freq,London_Freq,NewYork_Freq,Frankfurt_Freq,Toronto_Freq])

4.3 Evaluation

Compute pairwise correlation of cities according to venue category frequency values by pd.DataFrame.corr().

And the answer is :

	Shanghai	London	NewYork	Frankfurt	Toronto
Shanghai	1.000000	0.636168	0.558532	0.480385	0.729699
London	0.636168	1.000000	0.637411	0.611683	0.791106
NewYork	0.558532	0.637411	1.000000	0.544476	0.727447
Frankfurt	0.480385	0.611683	0.544476	1.000000	0.574084
Toronto	0.729699	0.791106	0.727447	0.574084	1.000000

Plotting a heatmap correlation matrix by sns.heatmap as follows,

sns.heatmap(city_corr,annot=True, fmt=".3f",cmap="YlGnBu")

And the figure is shown as Fig.2

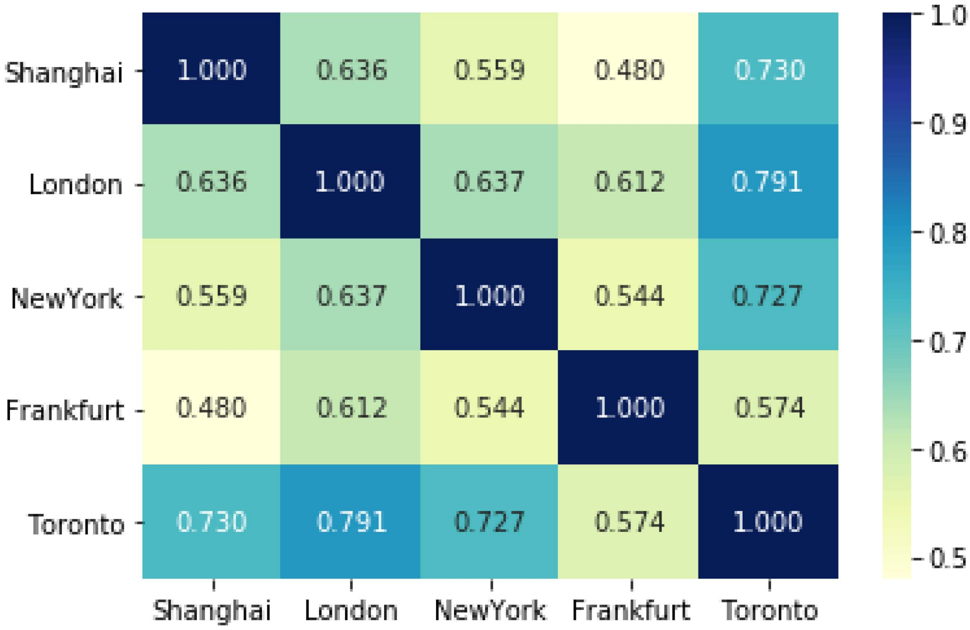

Fig.2

Chapter 5. Results

According to *city_corr*,we can find that the similarity of the 5 cities in descending order is:

1. Toronto-London: 0.791 （Correlation coefficient）
2. Toronto-Shanghai:0.730
3. Toronto-NewYork :0.727
4. NewYork-London: 0.637
5. London-Shanghai:0.636
6. Frankfurt-London:0.612
7. Toronto-Frankfurt:0.574
8. NewYork-Shanghai :0.559
9. Frankfurt-NewYork: 0.544
10. Frankfurt-Shanghai: 0.480

Chapter 6. Discussion

Let's take a look at Toronto-London venue frequency dataset to explore something interesting.

The top ten most frequent venues in Toronto is shown as Fig.3.

	coefficient
Coffee Shop	0.085579
Café	0.043972
Restaurant	0.031678
Park	0.025059
Bakery	0.022222
Pizza Place	0.021749
Italian Restaurant	0.021277
Sandwich Place	0.020331
Japanese Restaurant	0.019858
Hotel	0.018440

Fig.3

The top ten most frequent venues in London is shown as Fig.4.

	coefficient
Pub	0.066492
Coffee Shop	0.063575
Café	0.054826
Grocery Store	0.040537
Hotel	0.031788
Italian Restaurant	0.024205
Pizza Place	0.023914
Gym / Fitness Center	0.022747
Sandwich Place	0.021289
Bakery	0.020122

Fig.4

Now, we can intuitively see that the most frequent venues in these two cities are highly similar . This reflects the high similarity in the working environment, lifestyle and entertainment styles of the two cities

Chapter 7. Conclusion

For given metropolitan cities,I explored the similarity of them from the perspective of venue distribution. After web crawling, cleaning, feature acquisition and data integration, I obtain the location distribution and category of the venue in each district of a city. I calculated the frequency of the venue category in different cities and generated a correlation matrix between cities.

I prefer to think that the higher correlation coefficient of the venue occur frequency represents the higher similarity in the working environment, lifestyle and entertainment styles of the two cities.Therefore,for these five cities I explored in this assignment, **Toronto，Canada and London，the United Kingdom are the most similar, Frankfurt,Germany and Shanghai,China are the least similar.** For other cities in the world,I think my assignment can still be used as an effective solution for calculating city similarity.