Battle of neighborhoods

August 28, 2020

Capstone Project - The Battle of Neighborhoods

Introduction/Business problem:

What are the most common venues in World's major Central business districts (CBDs)? The answer to this question may be of interests to market researchers who will in turn provide the results to businesses who are intereted in finding out the most popular businesses to set up in CBDs in the world.

Description of the data:

In order to answer this question, we will explore the venue data from Foursquare for 3 World's CBDs and compare to find the most common venues:

Raffles Place (Singapore), Midtown (Manhattan, New York), Canary Wharf (London)

Methodology:

1. Utlised the venue data (venue name and venue categories) from Foursquare for 3 CBDs and use python "intersection" function to seek for the common subset of the venue categories data. The common subset will be the most common venues amongst the 3 CBDs.
2. Import Wordcloud to generate visualization of most popular venues based on frequency that each venue category appears in the merged list of venue data for all 3 CBDs. The more prominent the name of the venue appears in the Wordcloud image generated, the more popular the venue is.

[2]:
```
CLIENT_ID = '4JCOYMDTLXOR3IOVIGOEKSOQUDROEKY24UKCIWSQKOXCCYB2' # your␣
↪Foursquare ID
CLIENT_SECRET = 'LOZCARO4GP1DRDWZR1RJROFU2QMG1RTGUY2RI1Q3POS3AGVG' # your␣
↪Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: 4JCOYMDTLXOR3IOVIGOEKSOQUDROEKY24UKCIWSQKOXCCYB2
CLIENT_SECRET:LOZCARO4GP1DRDWZR1RJROFU2QMG1RTGUY2RI1Q3POS3AGVG
```

[3]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't␣
 ↪completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and␣
 ↪longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas␣
 ↪dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you␣
 ↪haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.3
  latest version: 4.8.4

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: /home/jupyterlab/conda/envs/python
```

```
  added / updated specs:
    - geopy


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    geographiclib-1.50         |             py_0          34 KB  conda-forge
    geopy-2.0.0                |     pyh9f0ad1d_0          63 KB  conda-forge
    openssl-1.1.1g             |       h516909a_1         2.1 MB  conda-forge
    ------------------------------------------------------------
                                           Total:         2.2 MB

The following NEW packages will be INSTALLED:

  geographiclib      conda-forge/noarch::geographiclib-1.50-py_0
  geopy              conda-forge/noarch::geopy-2.0.0-pyh9f0ad1d_0

The following packages will be UPDATED:

  openssl                                1.1.1g-h516909a_0 -->
1.1.1g-h516909a_1



Downloading and Extracting Packages
openssl-1.1.1g       | 2.1 MB    | ##################################### | 100%
geopy-2.0.0          | 63 KB     | ##################################### | 100%
geographiclib-1.50   | 34 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Libraries imported.
```

[4]:
```
#Singapore Central business district is Raffles Place
lat=1.2837
long=103.8509
#London Central business district is Canary Wharf
lat1=51.5061
long1=-0.0158
#New York Central business district in Manhattan is Midtown
lat2=40.754932
long2=-73.984016
```

[5]:
```
LIMIT = 100 # limit of number of venues returned by Foursquare API
radius = 500 # define radius
url = 'https://api.foursquare.com/v2/venues/explore?
 ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    lat,
    long,
    radius,
    LIMIT)
url # display URL
```

[5]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=4JCOYMDTLXOR3IOVIG0EKSO
QUDROEKY24UKCIWSQK0XCCYB2&client_secret=L0ZCARO4GP1DRDWZR1RJR0FU2QMG1RTGUY2RI1Q3
P0S3AGVG&v=20180605&ll=1.2837,103.8509&radius=500&limit=100'
```

[6]:
```
url1 = 'https://api.foursquare.com/v2/venues/explore?
 ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    lat1,
    long1,
    radius,
    LIMIT)
url1 # display URL
```

[6]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=4JCOYMDTLXOR3IOVIG0EKSO
QUDROEKY24UKCIWSQK0XCCYB2&client_secret=L0ZCARO4GP1DRDWZR1RJR0FU2QMG1RTGUY2RI1Q3
P0S3AGVG&v=20180605&ll=51.5061,-0.0158&radius=500&limit=100'
```

[7]:
```
url2 = 'https://api.foursquare.com/v2/venues/explore?
 ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    lat2,
    long2,
    radius,
    LIMIT)
url2 # display URL
```

[7]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=4JCOYMDTLXOR3IOVIG0EKSO
QUDROEKY24UKCIWSQK0XCCYB2&client_secret=L0ZCARO4GP1DRDWZR1RJR0FU2QMG1RTGUY2RI1Q3
P0S3AGVG&v=20180605&ll=40.754932,-73.984016&radius=500&limit=100'
```

[8]:
```
results = requests.get(url).json()
results1 = requests.get(url1).json()
results2 = requests.get(url2).json()
```

[9]:
```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

Singapore CBD-Rafles Place's list of venues

[10]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',␣
 ↪'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type,␣
 ↪axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:3: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[10]:

	name	categories	lat	lng
0	City Hot Pot	Hotpot Restaurant	1.284173	103.851585
1	CULINARYON	Comfort Food Restaurant	1.284876	103.850933
2	Virgin Active	Gym / Fitness Center	1.284608	103.850815
3	Fat Saigon Boy	Vietnamese Restaurant	1.282977	103.849068

```
4  The Salad Shop              Salad Place  1.285523  103.851177
```

[11]:
```
Raffles_Place_list=list(nearby_venues['categories'].unique())
Raffles_Place_list
```

[11]:
```
['Hotpot Restaurant',
 'Comfort Food Restaurant',
 'Gym / Fitness Center',
 'Vietnamese Restaurant',
 'Salad Place',
 'Hotel',
 'Cocktail Bar',
 'Modern European Restaurant',
 'Lounge',
 'Japanese Restaurant',
 'Waterfront',
 'Bar',
 'Café',
 'Gym',
 'Bridge',
 'Chinese Restaurant',
 'Martial Arts School',
 'Italian Restaurant',
 'Seafood Restaurant',
 'Yoga Studio',
 'Korean Restaurant',
 'French Restaurant',
 'Building',
 'Massage Studio',
 'Street Food Gathering',
 'Beer Garden',
 'Mexican Restaurant',
 'Plaza',
 'History Museum',
 'Sandwich Place',
 'Wine Bar',
 'Restaurant',
 'Food Court',
 'Coffee Shop',
 'Hotel Bar',
 'Nightclub',
 'Canal',
 'Cupcake Shop',
 'Speakeasy',
 'Australian Restaurant',
 'Pub',
 'Dumpling Restaurant',
```

```
 'Peking Duck Restaurant',
 'Asian Restaurant',
 'Greek Restaurant',
 'Szechuan Restaurant',
 'Cantonese Restaurant',
 'Temple']
```

London CBD-Canary Wharf's list of venues

[12]:
```
venues1 = results1['response']['groups'][0]['items']

nearby_venues1 = json_normalize(venues1) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',␣
 ↪'venue.location.lng']
nearby_venues1 =nearby_venues1.loc[:, filtered_columns]

# filter the category for each row
nearby_venues1['venue.categories'] = nearby_venues1.apply(get_category_type,␣
 ↪axis=1)

# clean columns
nearby_venues1.columns = [col.split(".")[-1] for col in nearby_venues1.columns]

nearby_venues1.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:3: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[12]:

	name	categories	lat	lng
0	Billingsgate Market	Fish Market	51.506312	-0.014184
1	Crossrail Place Roof Garden	Garden	51.505965	-0.016822
2	Sticks'n'Sushi	Sushi Restaurant	51.506178	-0.018287
3	Ippudo	Ramen Restaurant	51.505982	-0.018141
4	BrewDog Canary Wharf	Beer Bar	51.504447	-0.015185

[13]:
```
Canary_Wharf_list=list(nearby_venues1['categories'].unique())
Canary_Wharf_list
```

[13]:
```
['Fish Market',
 'Garden',
 'Sushi Restaurant',
 'Ramen Restaurant',
```

```
'Beer Bar',
'Supermarket',
'Movie Theater',
'Speakeasy',
'Shopping Mall',
'Cycle Studio',
'Gym / Fitness Center',
'Japanese Restaurant',
'Poke Place',
'Indian Restaurant',
'Pizza Place',
'Coffee Shop',
'Bakery',
'Mexican Restaurant',
'Hotel',
'Park',
'English Restaurant',
'Italian Restaurant',
'Gift Shop',
'Juice Bar',
'Steakhouse',
'Smoothie Shop',
'Plaza',
'Spanish Restaurant',
'Street Food Gathering',
'Fried Chicken Joint',
'Scenic Lookout',
'Burger Joint',
'Donut Shop',
'Sandwich Place',
'Bar',
'Salad Place',
'Breakfast Spot',
'Jewelry Store',
'Dumpling Restaurant',
'Chocolate Shop',
'Hotel Bar',
'Tennis Court',
'Bookstore',
'Vietnamese Restaurant',
'Deli / Bodega',
'Falafel Restaurant',
'Restaurant',
'Fast Food Restaurant',
'Café',
'Asian Restaurant',
'Portuguese Restaurant',
```

```
 'Greek Restaurant',
 'Stationery Store',
 'French Restaurant',
 'Pharmacy',
 'Grocery Store',
 'Athletics & Sports',
 'Bistro',
 'Outdoor Sculpture',
 'Harbor / Marina']
```

Manhattan, New York-CBD Midtown's list of venues

[14]:
```
venues2 = results2['response']['groups'][0]['items']

nearby_venues2 = json_normalize(venues2) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',␣
 ↪'venue.location.lng']
nearby_venues2 =nearby_venues2.loc[:, filtered_columns]

# filter the category for each row
nearby_venues2['venue.categories'] = nearby_venues2.apply(get_category_type,␣
 ↪axis=1)

# clean columns
nearby_venues2.columns = [col.split(".")[-1] for col in nearby_venues2.columns]

nearby_venues2.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:3: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[14]:

	name	categories	lat	lng
0	Bryant Park	Park	40.753621	-73.983265
1	Books Kinokuniya	Bookstore	40.754053	-73.984649
2	Belasco Theatre	Theater	40.756548	-73.983923
3	Blue Bottle Coffee	Coffee Shop	40.753027	-73.984140
4	sweetgreen	Salad Place	40.754640	-73.983102

[15]:
```
Midtown_list=list(nearby_venues2['categories'].unique())
Midtown_list
```

```
[15]: ['Park',
       'Bookstore',
       'Theater',
       'Coffee Shop',
       'Salad Place',
       'Grocery Store',
       'French Restaurant',
       'Mediterranean Restaurant',
       'Hotel',
       'Japanese Restaurant',
       'Gym',
       'Cycle Studio',
       'Pharmacy',
       'Bakery',
       'Café',
       'Pizza Place',
       'Cosmetics Shop',
       'Burrito Place',
       'Historic Site',
       'Chinese Restaurant',
       'Plaza',
       'Pilates Studio',
       'Sushi Restaurant',
       'Shoe Store',
       'Concert Hall',
       'Indian Restaurant',
       'Steakhouse',
       'Deli / Bodega',
       'Clothing Store',
       'Bar',
       'Comic Shop',
       'American Restaurant',
       'Miscellaneous Shop',
       'Szechuan Restaurant',
       'Juice Bar',
       'Mexican Restaurant',
       'Spa',
       'Cuban Restaurant',
       'Yoga Studio',
       'Smoke Shop',
       'Burger Joint',
       'Food Stand',
       'Art Gallery',
       'Tailor Shop',
       'Lounge',
       'Taco Place',
       'Office',
```

```
 'South American Restaurant',
 'Fast Food Restaurant',
 'Theme Restaurant',
 'Seafood Restaurant',
 'BBQ Joint',
 'Martial Arts School',
 'Arts & Crafts Store',
 'Sporting Goods Shop',
 'Exhibit',
 'Music Store',
 'Sandwich Place',
 'Gourmet Shop']
```

[16]:
```
common1=list(set(Raffles_Place_list).intersection(Midtown_list))
common1
```

[16]:
```
['Lounge',
 'Sandwich Place',
 'Plaza',
 'Gym',
 'Hotel',
 'Bar',
 'Yoga Studio',
 'Szechuan Restaurant',
 'Café',
 'French Restaurant',
 'Coffee Shop',
 'Salad Place',
 'Chinese Restaurant',
 'Seafood Restaurant',
 'Martial Arts School',
 'Mexican Restaurant',
 'Japanese Restaurant']
```

[17]:
```
common2=list(set(Raffles_Place_list).intersection(Canary_Wharf_list))
common2
```

[17]:
```
['Dumpling Restaurant',
 'Plaza',
 'Bar',
 'Hotel',
 'Street Food Gathering',
 'Gym / Fitness Center',
 'Speakeasy',
 'Vietnamese Restaurant',
 'Café',
 'Greek Restaurant',
```

```
 'French Restaurant',
 'Japanese Restaurant',
 'Italian Restaurant',
 'Hotel Bar',
 'Restaurant',
 'Coffee Shop',
 'Salad Place',
 'Asian Restaurant',
 'Sandwich Place',
 'Mexican Restaurant']
```

[18]:
```
common3=list(set(Midtown_list).intersection(Canary_Wharf_list))
common3
```

[18]:
```
['Bar',
 'Bakery',
 'Juice Bar',
 'Sandwich Place',
 'Sushi Restaurant',
 'Japanese Restaurant',
 'Park',
 'French Restaurant',
 'Pharmacy',
 'Pizza Place',
 'Deli / Bodega',
 'Coffee Shop',
 'Salad Place',
 'Mexican Restaurant',
 'Plaza',
 'Hotel',
 'Café',
 'Bookstore',
 'Steakhouse',
 'Cycle Studio',
 'Burger Joint',
 'Grocery Store',
 'Fast Food Restaurant',
 'Indian Restaurant']
```

Most common venues in 3 major Central business districts (CBDs) in the World are:

[19]:
```
common_venues=list(set(common1).intersection(common2))
common_venues
```

[19]:
```
['Plaza',
 'Bar',
 'Hotel',
```

```
 'Café',
 'French Restaurant',
 'Coffee Shop',
 'Salad Place',
 'Sandwich Place',
 'Mexican Restaurant',
 'Japanese Restaurant']
```

Most popular venues in terms of frequency?

[20]:
```
# install wordcloud
!conda install -c conda-forge wordcloud==1.4.1 --yes

print ('Wordcloud is installed and imported!')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible
solve.
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.3
  latest version: 4.8.4

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: /home/jupyterlab/conda/envs/python

  added / updated specs:
    - wordcloud==1.4.1


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    wordcloud-1.4.1            |           py36_0         324 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         324 KB
```

```
The following NEW packages will be INSTALLED:

  wordcloud          conda-forge/linux-64::wordcloud-1.4.1-py36_0


Downloading and Extracting Packages
wordcloud-1.4.1      | 324 KB    | ###################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Wordcloud is installed and imported!
```

[40]:
```
# import package
from wordcloud import WordCloud

%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

mpl.style.use('ggplot') # optional: for ggplot-like style

# check for latest version of Matplotlib
print ('Matplotlib version: ', mpl.__version__) # >= 2.0.0
```

```
Matplotlib version:  3.3.0
```

[21]:
```
all_venues=pd.concat([nearby_venues,nearby_venues1,nearby_venues2])
all_venues.head()
```

[21]:

	name	categories	lat	lng
0	City Hot Pot	Hotpot Restaurant	1.284173	103.851585
1	CULINARYON	Comfort Food Restaurant	1.284876	103.850933
2	Virgin Active	Gym / Fitness Center	1.284608	103.850815
3	Fat Saigon Boy	Vietnamese Restaurant	1.282977	103.849068
4	The Salad Shop	Salad Place	1.285523	103.851177

[22]:
```
count=pd.DataFrame(all_venues['categories'].value_counts())
count.head()
```

[22]:

	categories
Coffee Shop	17
Hotel	13
Theater	9
Café	7
Sandwich Place	6

[23]:
```
count1=count.reset_index()
count1.head()
```

[23]:

	index	categories
0	Coffee Shop	17
1	Hotel	13
2	Theater	9
3	Café	7
4	Sandwich Place	6

[24]:
```
count1.columns=['categories','count']
count1.head()
```

[24]:

	categories	count
0	Coffee Shop	17
1	Hotel	13
2	Theater	9
3	Café	7
4	Sandwich Place	6

[25]:
```
count1['count'].sum()
```

[25]:
```
264
```

[33]:
```
max_words = 90
word_string = '*'
for i in count1.index.values:
        repeat_num_times = int(count1.loc[i, 'count']/264*max_words)
        word_string = word_string + ((count1['categories'] + '*') *␣
 ↪repeat_num_times)

# display the generated text
word_string
```

[33]:
```
0      *Coffee Shop*Coffee Shop*Coffee Shop*Coffee Sh…
1      *Hotel*Hotel*Hotel*Hotel*Hotel*Hotel*Hotel*Hot…
2      *Theater*Theater*Theater*Theater*Theater*Theat…
3      *Café*Café*Café*Café*Café*Café*Café*Café*Café*…
4      *Sandwich Place*Sandwich Place*Sandwich Place*…
5      *Japanese Restaurant*Japanese Restaurant*Japan…
6      *Plaza*Plaza*Plaza*Plaza*Plaza*Plaza*Plaza*Pla…
7      *Gym / Fitness Center*Gym / Fitness Center*Gym…
8      *Sushi Restaurant*Sushi Restaurant*Sushi Resta…
9      *Park*Park*Park*Park*Park*Park*Park*Park*Park*…
10     *French Restaurant*French Restaurant*French Re…
11     *Bookstore*Bookstore*Bookstore*Bookstore*Books…
12     *Salad Place*Salad Place*Salad Place*Salad Pla…
```

```
13      *Italian Restaurant*Italian Restaurant*Italian…
14      *Bakery*Bakery*Bakery*Bakery*Bakery*Bakery*Bak…
15      *Gym*Gym*Gym*Gym*Gym*Gym*Gym*Gym*Gym*Gym*Gym*G…
16      *Clothing Store*Clothing Store*Clothing Store*…
17      *Burger Joint*Burger Joint*Burger Joint*Burger…
18      *Seafood Restaurant*Seafood Restaurant*Seafood…
19      *Cocktail Bar*Cocktail Bar*Cocktail Bar*Cockta…
20      *Mexican Restaurant*Mexican Restaurant*Mexican…
21      *Fast Food Restaurant*Fast Food Restaurant*Fas…
22      *Cuban Restaurant*Cuban Restaurant*Cuban Resta…
23      *Cycle Studio*Cycle Studio*Cycle Studio*Cycle …
24      *Bar*Bar*Bar*Bar*Bar*Bar*Bar*Bar*Bar*Bar*Bar*B…
25      *Lounge*Lounge*Lounge*Lounge*Lounge*Lounge*Lou…
26      *Juice Bar*Juice Bar*Juice Bar*Juice Bar*Juice…
27      *Hotel Bar*Hotel Bar*Hotel Bar*Hotel Bar*Hotel…
28      *Stationery Store*Stationery Store*Stationery …
29      *Martial Arts School*Martial Arts School*Marti…
30      *Speakeasy*Speakeasy*Speakeasy*Speakeasy*Speak…
31      *Wine Bar*Wine Bar*Wine Bar*Wine Bar*Wine Bar*…
32      *Grocery Store*Grocery Store*Grocery Store*Gro…
33      *Pharmacy*Pharmacy*Pharmacy*Pharmacy*Pharmacy*…
34      *Nightclub*Nightclub*Nightclub*Nightclub*Night…
35      *Deli / Bodega*Deli / Bodega*Deli / Bodega*Del…
36      *Food Court*Food Court*Food Court*Food Court*F…
37      *Vietnamese Restaurant*Vietnamese Restaurant*V…
38      *Asian Restaurant*Asian Restaurant*Asian Resta…
39      *Cosmetics Shop*Cosmetics Shop*Cosmetics Shop*…
40      *Gift Shop*Gift Shop*Gift Shop*Gift Shop*Gift …
41      *Korean Restaurant*Korean Restaurant*Korean Re…
42      *Waterfront*Waterfront*Waterfront*Waterfront*W…
43      *Chinese Restaurant*Chinese Restaurant*Chinese…
44      *Yoga Studio*Yoga Studio*Yoga Studio*Yoga Stud…
45      *Szechuan Restaurant*Szechuan Restaurant*Szech…
46      *Shopping Mall*Shopping Mall*Shopping Mall*Sho…
47      *Modern European Restaurant*Modern European Re…
48      *Concert Hall*Concert Hall*Concert Hall*Concer…
49      *Indian Restaurant*Indian Restaurant*Indian Re…
50      *Greek Restaurant*Greek Restaurant*Greek Resta…
51      *Dumpling Restaurant*Dumpling Restaurant*Dumpl…
52      *Street Food Gathering*Street Food Gathering*S…
53      *Restaurant*Restaurant*Restaurant*Restaurant*R…
54      *Pizza Place*Pizza Place*Pizza Place*Pizza Pla…
55      *Steakhouse*Steakhouse*Steakhouse*Steakhouse*S…
56      *Harbor / Marina*Harbor / Marina*Harbor / Mari…
57      *Australian Restaurant*Australian Restaurant*A…
58      *Bistro*Bistro*Bistro*Bistro*Bistro*Bistro*Bis…
59      *Pub*Pub*Pub*Pub*Pub*Pub*Pub*Pub*Pub*Pub*Pub*P…
```

```
60      *Food Stand*Food Stand*Food Stand*Food Stand*F…
61      *Building*Building*Building*Building*Building*…
62      *Chocolate Shop*Chocolate Shop*Chocolate Shop*…
63      *Tailor Shop*Tailor Shop*Tailor Shop*Tailor Sh…
64      *Smoke Shop*Smoke Shop*Smoke Shop*Smoke Shop*S…
65      *Spa*Spa*Spa*Spa*Spa*Spa*Spa*Spa*Spa*Spa*Spa*S…
66      *Fried Chicken Joint*Fried Chicken Joint*Fried…
67      *Cantonese Restaurant*Cantonese Restaurant*Can…
68      *Canal*Canal*Canal*Canal*Canal*Canal*Canal*Can…
69      *Breakfast Spot*Breakfast Spot*Breakfast Spot*…
70      *Comfort Food Restaurant*Comfort Food Restaura…
71      *Peking Duck Restaurant*Peking Duck Restaurant…
72      *Movie Theater*Movie Theater*Movie Theater*Mov…
73      *American Restaurant*American Restaurant*Ameri…
74      *Taco Place*Taco Place*Taco Place*Taco Place*T…
75      *Mediterranean Restaurant*Mediterranean Restau…
76      *Miscellaneous Shop*Miscellaneous Shop*Miscell…
77      *Spanish Restaurant*Spanish Restaurant*Spanish…
78      *Supermarket*Supermarket*Supermarket*Supermark…
79      *Jewelry Store*Jewelry Store*Jewelry Store*Jew…
80      *Massage Studio*Massage Studio*Massage Studio*…
81      *Smoothie Shop*Smoothie Shop*Smoothie Shop*Smo…
82      *Exhibit*Exhibit*Exhibit*Exhibit*Exhibit*Exhib…
83      *Historic Site*Historic Site*Historic Site*His…
84      *Gourmet Shop*Gourmet Shop*Gourmet Shop*Gourme…
85      *Outdoor Sculpture*Outdoor Sculpture*Outdoor S…
86      *English Restaurant*English Restaurant*English…
87      *Athletics & Sports*Athletics & Sports*Athleti…
88      *Comic Shop*Comic Shop*Comic Shop*Comic Shop*C…
89      *Beer Garden*Beer Garden*Beer Garden*Beer Gard…
90      *Falafel Restaurant*Falafel Restaurant*Falafel…
91      *Cupcake Shop*Cupcake Shop*Cupcake Shop*Cupcak…
92      *Burrito Place*Burrito Place*Burrito Place*Bur…
93      *South American Restaurant*South American Rest…
94      *History Museum*History Museum*History Museum*…
95      *Office*Office*Office*Office*Office*Office*Off…
96      *Fish Market*Fish Market*Fish Market*Fish Mark…
97      *Donut Shop*Donut Shop*Donut Shop*Donut Shop*D…
98      *Temple*Temple*Temple*Temple*Temple*Temple*Tem…
99      *Beer Bar*Beer Bar*Beer Bar*Beer Bar*Beer Bar*…
100     *Scenic Lookout*Scenic Lookout*Scenic Lookout*…
101     *BBQ Joint*BBQ Joint*BBQ Joint*BBQ Joint*BBQ J…
102     *Arts & Crafts Store*Arts & Crafts Store*Arts …
103     *Tennis Court*Tennis Court*Tennis Court*Tennis…
104     *Garden*Garden*Garden*Garden*Garden*Garden*Gar…
105     *Theme Restaurant*Theme Restaurant*Theme Resta…
106     *Shoe Store*Shoe Store*Shoe Store*Shoe Store*S…
```

```
107    *Pilates Studio*Pilates Studio*Pilates Studio*…
108    *Poke Place*Poke Place*Poke Place*Poke Place*P…
109    *Art Gallery*Art Gallery*Art Gallery*Art Galle…
110    *Music Store*Music Store*Music Store*Music Sto…
111    *Ramen Restaurant*Ramen Restaurant*Ramen Resta…
112    *Portuguese Restaurant*Portuguese Restaurant*P…
113    *Bridge*Bridge*Bridge*Bridge*Bridge*Bridge*Bri…
114    *Hotpot Restaurant*Hotpot Restaurant*Hotpot Re…
115    *Sporting Goods Shop*Sporting Goods Shop*Sport…
Name: categories, dtype: object
```

[38]:
```
# create the word cloud
wc = WordCloud(
    background_color='white',
    max_words=2000
)
wc = WordCloud(background_color='white').generate(str(word_string))

print('Word cloud created!')
```

```
Word cloud created!
```

[43]:
```
fig = plt.figure()
fig.set_figwidth(14) # set width
fig.set_figheight(18) # set height

# display the word cloud
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Observation and Conclusion:

Based on the results of most common venues from the 3 CBDs, we can have a few recommendations for potential business ideas in CBDs:

Risk-averse business owners may be more keen to opening businesses that have been proven to be popular in CBDs as shown in the list of most common venues above. Risk-lovers, in opposite, may prefer to open businesses that are not yet too common and thus may want to introduce something more unique.

[ ]: