An approach for cities clustering

Phuc Tran Duy

April 15, 2019

1. Introduction

1.1. Background

One of the most concern problems of international companies when entering new market is understanding market's culture. However, they can apply their success case studies of similar market in order to reduce analytic and research effort for new market. To achieve this, they must have a method to measure how similar one city with others. This paper propose an approach to represent city by a vector base on nearby venues and then calculate similarity between them.

1.2. Problem

Using venue data of cities in a specific area which might include category tags for each venue to transform to a vector which represents a city. This project aim to tell how similarity between two cities, then apply to clustering a larger set of cities.

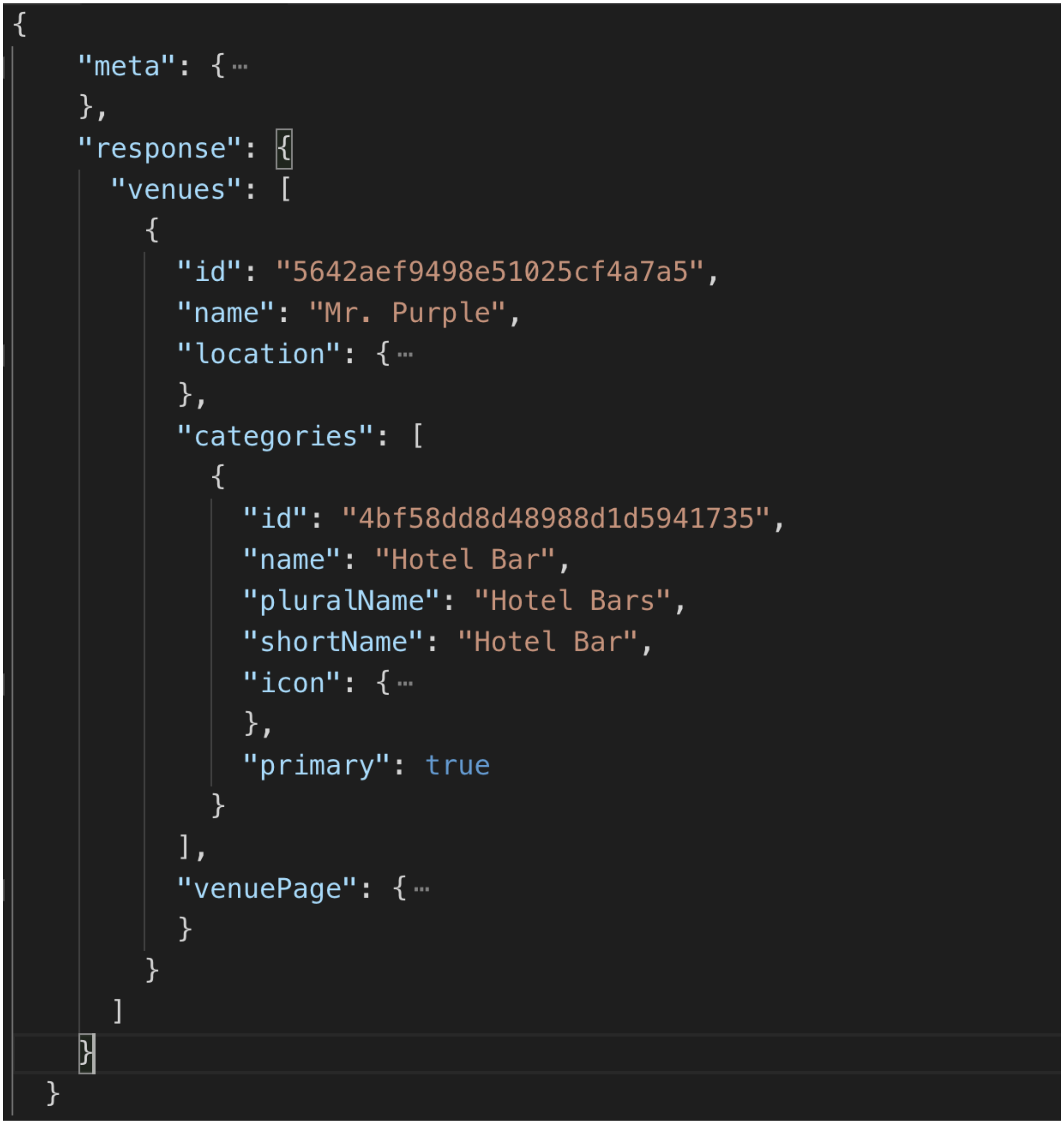
2. Data acquisition and cleaning

2.1. Data sources

In order to get geography location of each city, I use Google Map API. Fortunately, there are Python client library which easily integrate with code in notebook.

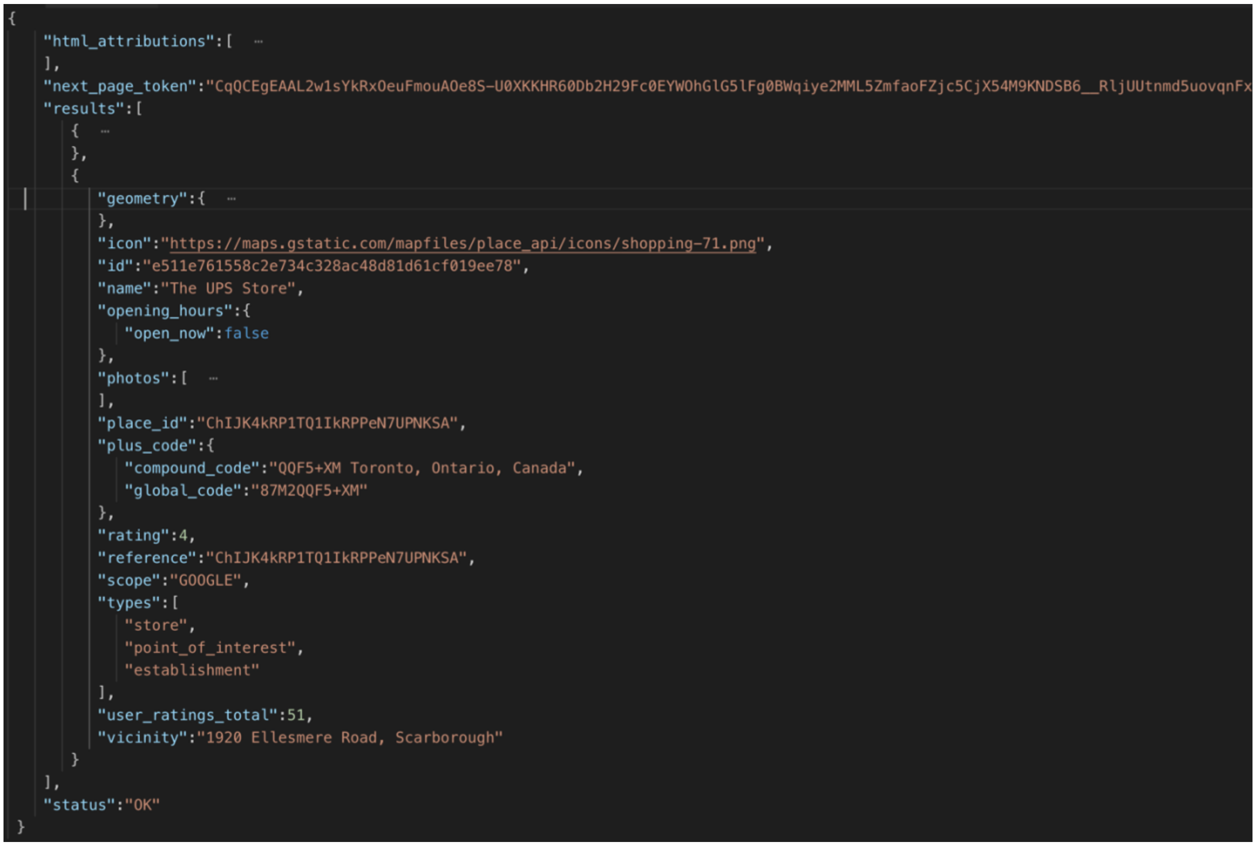
Base on geography location of each city, I can get nearby venues by using Foursquare API and Google Map API. All we need is venue category field in response from those API.

Example response from Foursquare API (we care about **categories** field):



```
{
    "meta": {…
    },
    "response": {
      "venues": [
        {
          "id": "5642aef9498e51025cf4a7a5",
          "name": "Mr. Purple",
          "location": {…
          },
          "categories": [
            {
              "id": "4bf58dd8d48988d1d5941735",
              "name": "Hotel Bar",
              "pluralName": "Hotel Bars",
              "shortName": "Hotel Bar",
              "icon": {…
              },
              "primary": true
            }
          ],
          "venuePage": {…
          }
        }
      ]
    }
  }
```

Example response from Google Map API (we care about **types** field):



```
{
    "html_attributions":[  ⋯
    ],
    "next_page_token":"CqQCEgEAAL2w1sYkRxOeuFmouAOe8S-U0XKKHR60Db2H29Fc0EYWOhGlG5lFg0BWqiye2MML5ZmfaoFZjc5CjX54M9KNDSB6__RljUUtnmd5uovqnFx
    "results":[
        {  ⋯
        },
        {
            "geometry":{  ⋯
            },
            "icon":"https://maps.gstatic.com/mapfiles/place_api/icons/shopping-71.png",
            "id":"e511e761558c2e734c328ac48d81d61cf019ee78",
            "name":"The UPS Store",
            "opening_hours":{
                "open_now":false
            },
            "photos":[  ⋯
            ],
            "place_id":"ChIJK4kRP1TQ1IkRPPeN7UPNKSA",
            "plus_code":{
                "compound_code":"QQF5+XM Toronto, Ontario, Canada",
                "global_code":"87M2QQF5+XM"
            },
            "rating":4,
            "reference":"ChIJK4kRP1TQ1IkRPPeN7UPNKSA",
            "scope":"GOOGLE",
            "types":[
                "store",
                "point_of_interest",
                "establishment"
            ],
            "user_ratings_total":51,
            "vicinity":"1920 Ellesmere Road, Scarborough"
        }
    ],
    "status":"OK"
}
```

Since venue classify convention is different between two API, we need a normalization stage.

2.2. Data cleaning