Google Trends API for Python

In this tutorial, I will demonstrate how to use the Google Tre
for getting the current trending topics on the internet.

Connect to a new runtime

Introduction

Google trends is a website that analyzes and lists the popular search results on Google search based on various regions and languages. Google Trends is Google's website (obviously). With the help of this tutorial, you can get the trending results and many more from google trends website using python. You don't need to manually search and copy the trending results, the Python API called `pytrends` does the job for you. Before getting started, I want all of you guys to go through the official documentation of the `pytrends` API.

pytrends API

Installation

The first step is to install the library manually. So, open your favorite IDE or notebook start typing the following code. I will use [Google Colab](https://colab.research.google.com) because it's my favorite notebook.

Connect to a new runtime

> If you are using jupyter notebook, just type the code as it is (make sure you have
> the beginning)

```
!pip install pytrends
```

```
Collecting pytrends
  Downloading https://files.pythonhosted.org/packages/74/a4/c1b1242be7d31650c6d9128a776
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (from py
Requirement already satisfied: lxml in /usr/local/lib/python3.6/dist-packages (from pytr
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from
Building wheels for collected packages: pytrends
  Building wheel for pytrends (setup.py) ... done
  Created wheel for pytrends: filename=pytrends-4.7.2-cp36-none-any.whl size=14261 sha25
  Stored in directory: /root/.cache/pip/wheels/64/ae/af/51d48fbbca0563036c6f80999b7ce3f6
Successfully built pytrends
Installing collected packages: pytrends
Successfully installed pytrends-4.7.2
```

Or, if you are using an IDE, just type the following code

```
pip install pytrends
```

After executing the above code you should get a successful message as shown above

Implementation

Connecting to Google

You must connect to Google first because after all, we are requesting the Google trending topics from Google Trends. For this, we need to import the method called `TrendReq` from `pytrends.request` library. Also, I will import the pandas library to store and visualize the data which you see in the later tutorial.

```
import pandas as pd
from pytrends.request import TrendReq
pytrend = TrendReq()
```

Connect to a new runtime

Interest By Region

Let us see the terms which are popular in the region worldwide. I will choose, the term to be searched as "Taylor Swift" (I like her so....).

```
pytrend.build_payload(kw_list=['Taylor Swift'])
# Interest by Region
df = pytrend.interest_by_region()
df.head(10)
```

	Taylor Swift
geoName	
Afghanistan	0
Albania	0
Algeria	16
American Samoa	0
Andorra	0
Angola	0
Anguilla	0
Antarctica	0
Antigua & Barbuda	0
Argentina	19

Now you might be thinking what are the values, what do they denote?

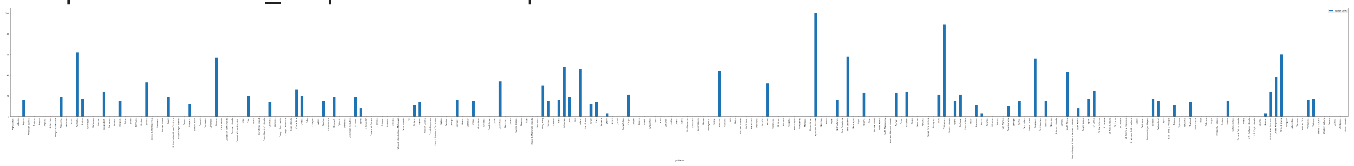
The values are calculated on a scale from 0 to 100, where 100 is the location with the most popularity as a fraction of total searches in that location, a value of 50 indicates a location which is half as popular. A value of 0 indicates a location where there was not enough data for this term. Source → Google Trends..

Let us plot the result on a bar graph because sometimes visual representation gives a clear picture.

```
df.reset_index().plot(x='geoName', y='Taylor Swift', figsize=(120, 10), kind =
```

Connect to a new runtime

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff024a40b70>
```



Also, you use the parameter `resolution = 'COUNTRY_NAME'` to filter the results.

Daily Search Trends

Now let us get the top daily search trends worldwide. To do this we have to use the `trending_searches()` method. If you want to search worldwide just don't pass any parameter.

```
# Get Google Hot Trends data
df = pytrend.trending_searches(pn='united_states')
df.head()
```

	0
0	South Carolina primary
1	David Byrne
2	Liverpool
3	Lakers vs Grizzlies
4	Tom Steyer

Make sure you enter the country name in lowercase `pn = "canada"` . Also, you can compare the above results with the google trend's result. To get today's trending topics just use:

```
df = pytrend.today_searches(pn='US')
```

Top Charts

Let us see what was trending in 2019. With the help of `top_charts` method we can get the top trending searches yearly.

Connect to a new runtime

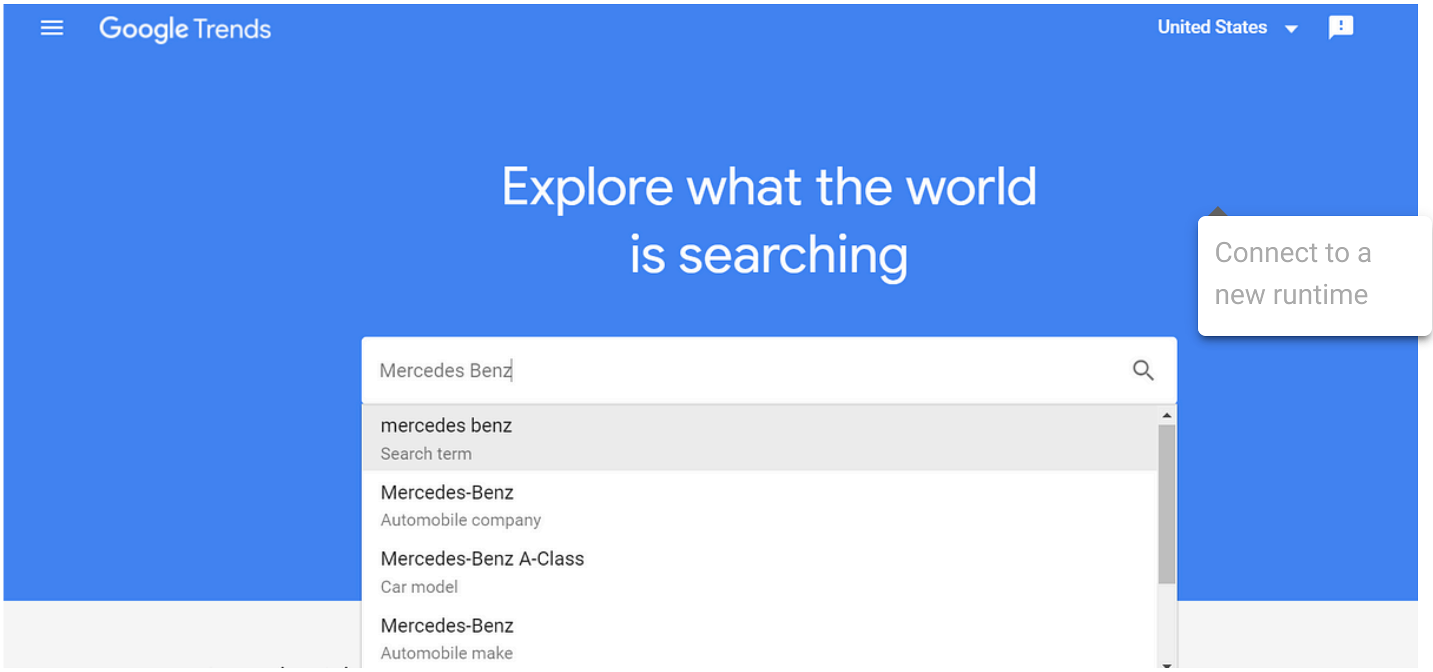
```
# Get Google Top Charts
df = pytrend.top_charts(2019, hl='en-US', tz=300, geo='GLOBAL')
df.head()
```

	title	exploreQuery
0	India vs South Africa	
1	Cameron Boyce	
2	Copa America	
3	Bangladesh vs India	
4	iPhone 11	

To compare the results just visit Google Trends. We can specify the year and the country that we want to see the trending searches.

Google Keyword Suggestions

Let us see how can we obtain google's keyword suggestion. If you don't know what I'm talking about. The below image explains things more clear.



```
# Get Google Keyword Suggestions
keywords = pytrend.suggestions(keyword='Mercedes Benz')
df = pd.DataFrame(keywords)
df.drop(columns= 'mid')   # This column makes no sense
```

	title	type
0	Mercedes-Benz	Automobile company
1	Mercedes-Benz A-Class	Car model
2	Mercedes-Benz	Automobile make
3	Mercedes-Benz E-Class	Car model
4	Mercedes-Benz GLB-Class	Car model

Related Queries

It's a common thing that when a user searches for a topic, they would also search for something related. These are called related queries. Let us see what are the related queries for the topic "***Coronavirus***". Always remember when you want to change the topic name just run the following code again with the new name as the parameter.

```
pytrend.build_payload(kw_list=['Coronavirus'])
```

Now let's run the method `related_queries` which returns a dictionary full of related queries for the topic ***Coronavirus***

```
# Related Queries, returns a dictionary of dataframes
related_queries = pytrend.related_queries()
related_queries.values()
```

Connect to a new runtime

```
dict_values([{'top':                       query  value
0                     virus    100
1         virus coronavirus     95
2                    corona     92
3         china coronavirus     87
4                     china     86
5      coronavirus symptoms     83
6          news coronavirus     72
7              corona virus     61
8        coronavirus update     53
9        coronavirus italia     50
10           el coronavirus     37
11          coronavirus map     34
12        wuhan coronavirus     33
13                    wuhan     33
14        coronavirus death     31
15      what is coronavirus     31
16        coronavirus cases     30
17          coronavirus usa     30
18     sintomas coronavirus     30
19           uk coronavirus     23
20           us coronavirus     23
21  symptoms of coronavirus     22
22       coronavirus latest     20
23         coronavirus live     20
24     coronavirus in china     20, 'rising':                          query   value
0            wuhan coronavirus  168350
1                        wuhan  165100
2          notizie coronavirus   71950
3           ultime coronavirus   64800
4   coronavirus ultime notizie   57900
5           milano coronavirus   43800
6        coronavirus lombardia   43450
7        coronavirus in italia   37300
8      wuhan china coronavirus   36700
9          italien coronavirus   28950
10            coronavirus roma   28600
11          coronavirus veneto   27750
12                 wuhan virus   26550
13        coronavirus map live   26100
14   coronavirus symptoms 2020   24900
15           mappa coronavirus   24850
16            coronavirus meme   23950
17           coronavirus count   23900
18      coronavirus death rate   22950
19   aggiornamenti coronavirus   22950
20         coronavirus muertos   22550
21                 kobe bryant   22100
22       latest on coronavirus   21000
23          coronavirus napoli   20750
24          coronavirus torino   18750}])
```

Similarly, you can also search for the related topics just run the below code to do so:

Connect to a new runtime

```
# Related Topics, returns a dictionary of dataframes
related_topic = pytrend.related_topics()
related_topic.values()
```

```
dict_values([{'rising':      value  ...            topic_type
0   175600  ...         City in China
1    47900  ...        Italian region
2    27350  ...           Cooperative
3    26650  ...        Italian region
4    18550  ...                 Topic
5    17800  ...        Italian region
6    17600  ...               Website
7    15150  ...                 Topic
8    13550  ...        Italian region
9    12450  ...                Animal
10   11000  ...                 Topic
11   10500  ...                 Topic
12    9950  ...      Water navigation
13    9400  ...                 Topic
14    2800  ...       Spoken language
15    2400  ...          Ethnic group
16    2250  ...  Country in East Asia
17    2100  ...     Country in Europe
18    1050  ...                 Topic
19     500  ...                 Topic

[20 rows x 6 columns], 'top':     value  ...            topic_type
0     100  ...                 Virus
1       8  ...                 Topic
2       5  ...  Country in East Asia
3       5  ...      Infectious agent
4       4  ...     Country in Europe
5       4  ...                 Topic
6       3  ...       Spoken language
7       3  ...                 Virus
8       3  ...          Ethnic group
9       2  ...         City in China
10      1  ...               Disease
11      1  ...               Disease
12      1  ...                 Topic
13      1  ...                 Topic
14      0  ...        Italian region
15      0  ...           Cooperative
16      0  ...        Italian region
17      0  ...                 Topic
18      0  ...        Italian region
19      0  ...                 Website
20      0  ...                   Topic
21      0  ...          Italian region
22      0  ...                  Animal
23      0  ...                   Topic
24      0  ...                   Topic

[25 rows x 7 columns]}])
```

Connect to a
new runtime

This is the end of the tutorial, I hope you guys have learned a thing or two. If you guys have any doubts regarding the tutorial let me know via the comment section. Although this is a short tutorial there is a lot to learn. Alright see you in my next tutorial, have a good day!!!