Project Name:

TweePyTrends

Group Members:

Jiaqi Yang - stephyang
Ruize Liu - liu10
Dominic Teo – dominicteo
Ya-Han Cheng – yahancheng

Project Overview

The goal of this project is to see how people on social media (Twitter) interact with a certain topic, then compare it with Google search trends to get a more complete picture on the topic and to evaluate how it differs from Twitter users. This is important as there is a pervasive perception of the closed-loop nature of Twitter. We utilize data visualization techniques to make our result more presentable and readable to non-technical users, such as policy makers and marketers, and thus they can easily keep track of their ongoing projects with real-time Twitter scraping result or compare the result with that of similar topics in the past on Twitter and Google.

In this project, we utilize visualization techniques such as word cloud, line chart of the interest/frequency over time, and table presenting the key opinion leaders of the topic. With the word cloud, users can see what words are related to the topic and how often they are mentioned. With the line chart of the interest/frequency over time, users can see the Twitter interest and search frequency of the topic over time. With the key opinion leaders chart, users can find out who has the greatest impact for the topic on Twitter.

Software Structure

There are 10 python files, 12 image files and 1 csv file in our project folder.

1) **Code files:**
   a) **Google Trends Functions**
      i) ***GoogleTrends.py***: uses the PyTrends module to return Google Search data based on a search word, start date and end date. Then uses the data to create visualizations on search interest over time, search interest of related topics, word cloud on related queries and chart showing 20 countries with the highest search interest.
   b) **Real-time Twitter Crawler Functions:**
      i) ***util.py***: utility functions used in crawler, data cleaning and processing.
      ii) ***crawler.py***: a crawler function retrieving data from Twitter with a certain hashtag and intended maximum attempts.
      iii) ***word.py***: plotting function that generates word cloud taking list of strings as input (also used in Google Trends and Twitterscrapper part)
      iv) ***network.py***: plotting function that creates a network plot with hashtags as nodes. Edges show relationships between hashtags and size of nodes indicates its frequency in crawled data.
      v) ***barplot.py***: plotting function that generates a barplot to show the top 6 mentioned hashtags (it supposed like trending hashtags)
      vi) ***go.py***: pull all together, crawl data from Twitter pages and generate images for display.
   c) **Twitterscrapper Functions:**
      i) ***tw_scraper.p***y: a scraper function based on twitterscraper module recursively collects historical tweets containing a certain hashtag within a smaller time period to increase the completeness of search.
      ii) ***tw_plot.py***: visualize tweets data, plotting word cloud, frequency chart, and top 10 key opinion leaders (KOL).
      iii) ***go2.py***: pull tw_scraper and tw_plot together, run in command line.
   d) **User Interface Functions:**
      i) ***GUI.py***: build a GUI which can let the user see previous scraping examples as well as insert their own word and returns the result. Connected with GoogleTrends.py, go.py and go2.py
2) **Image files**
   a) Sample Images: Images generated by pre-scraped data using key = “coronavirus”, start_date = “2019-12-02”, end_date = “2020-03-01” for illustration purpose.
   b) Error Message: error message image displayed when crawler gets frozen by website’s anti-crawler system.
3) **Csv files:**

   Pre-scraped data using package twitterscrapper with key = “coronavirus”, start_date = “2019-12-02”, end_date = “2020-03-01”

Expected and Actual Outcomes

Expected outcome:

> We intended to create an interactive user interface, which takes in a keyword of interested topic and a specific time period, then returns visualization results based on Google trends and Twitter data to give users an overview of relative activities on Google and Twitter regarding the topic.

Actual outcome:

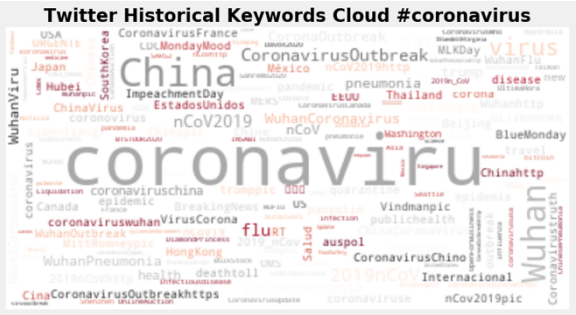
> Generally, we have finished what we want. There are just a few flaws for our software.

1) Unstable historical tweets scraping
   Because of the anti-crawler system on Twitter and the limitation with Twitter API, our request will get frozen or rejected after too many attempts, and we can crawl only historical data within 7 days with API. Our solution is to use a function based on Twitterscrapper, a GitHub module, to scrape historical tweets to avoid the limitation from the API and the anti-crawler system.
   However, the execution is unstable and takes a huge amount of time to scrape. Therefore, we have difficulties embedding the historical tweets scraping function in our GUI. User has to run go2.py in command line by themselves to get the initial csv file, then use GUI to see the visualization results.

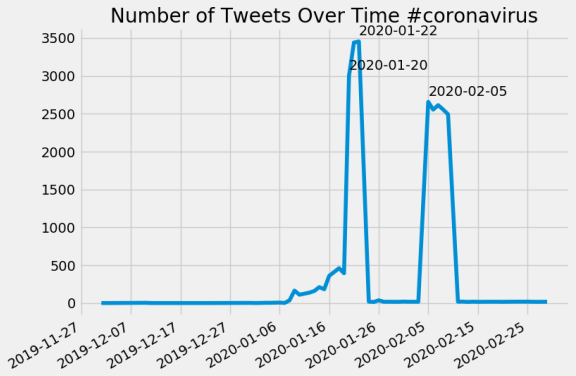
2) Some unexpected bugs for special searching words
   TweePyTrends works well for most words, but when searching for some special words like '2#@$s&$Er' (just a garble) or 'China', there may be some unexpected results because of too little scraping results and decoding issues

Appendix - Detailed Explanation of Our Design

1) Plot description
   a) Historical tweets plot:
      i) Word Cloud: a word cloud of hashtags which are used together in the same posts with the hashtag we search



      ii) Frequency chart: line chart of number of tweets over time with peaks highlighted

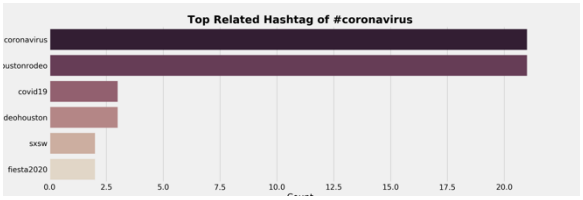


      iii) Top 10 KOL: Sort out top 10 key opinion leaders with highest popularity scores (popularity scores = 2*number of retweets + number of likes + number of replies)

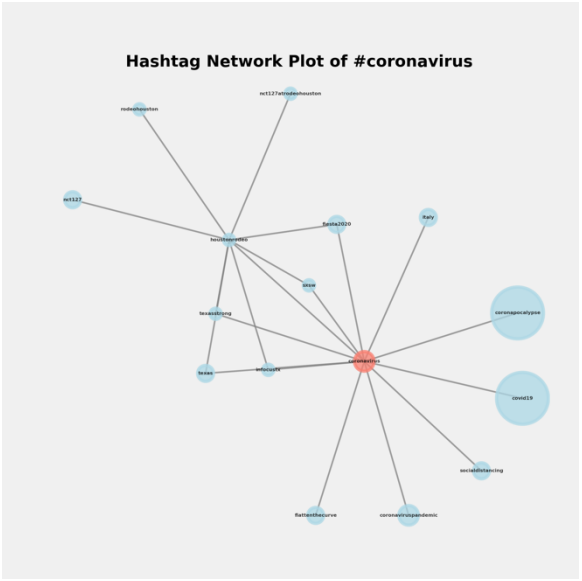
username	pop_score
World Health Organization (WHO)	40971.0
TY8	39112.0
Zulma Cucunubá	38677.0
Yikai Luo	25784.0
People's Daily, China	12313.0
Conflits	10589.0
Tom	9358.0
DW Español	6892.0
CDC	5816.0
Sara A. Carter	5546.0

b) Real-time twitter data:

i) Bar chart of top 5 related hashtags: see the current related hashtags

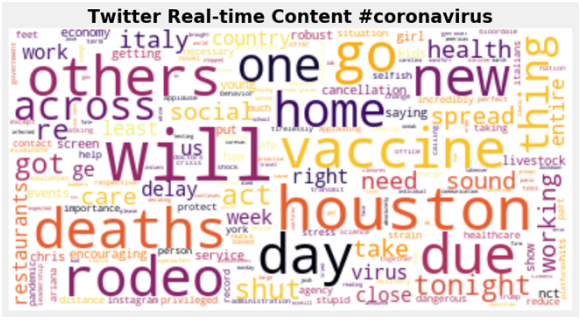


i) Hashtag network plot: network plot with hashtags as nodes, edges show relationships between hashtags and size of nodes indicates its frequency in crawled data



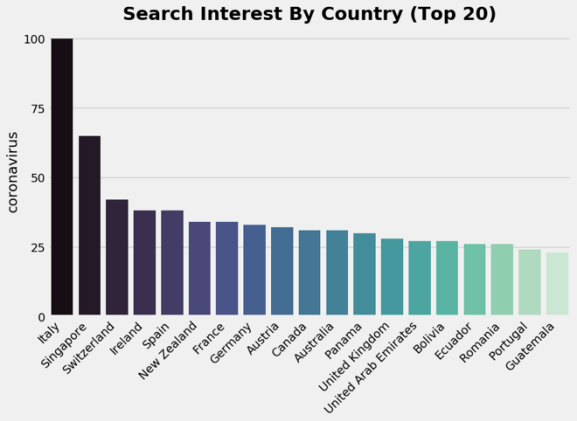
ii) Word cloud of related hashtags: a word cloud of hashtags which are used together in the same posts with the hashtag we search

iii) Word cloud of content: a word cloud of words in content which are used together in the same posts with the hashtag we search

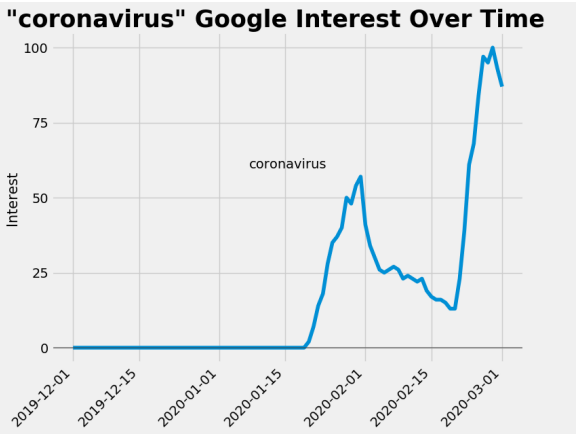


c) Google data from Pytrends:

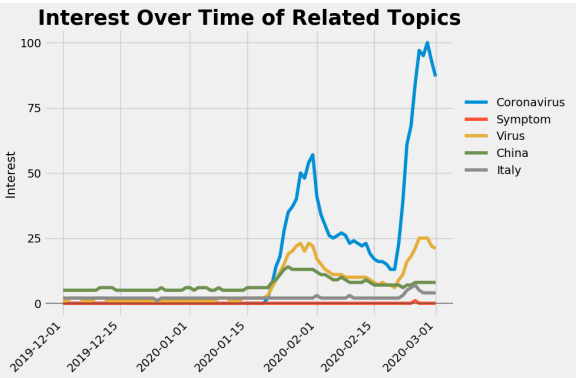
i) Bar chart of top 20 countries with highest search for our word:



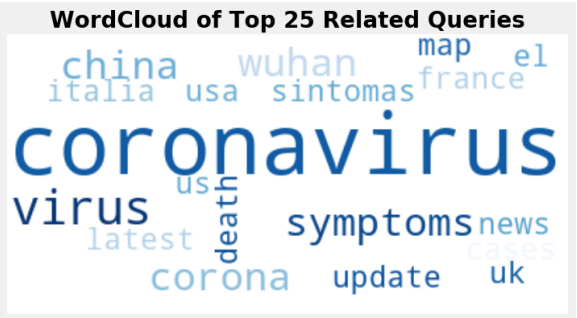
ii) Line chart of google interest of our word over time:



iii) Line chart of top 5 related topics:



iv) Word cloud of top 25 related queries:



2) Design details

a) Historical Twitter Data from *Twitterscraper* (API)

To avoid the constraints of the Twitter API (only 180 requests every 15 minutes), our Twitterscraper code is based on Twitterscraper, an online module from GitHub, which crawls tweets without the limitation but with an upper bound of the number of tweets we can crawl in every execution. As a result, we recursively scrape tweets by its hashtag with Twitterscraper in a small time period, trying to collect all public tweets in that time period.

There are three visualization plots with historical tweets data. One is the word cloud of hashtags which are used together with the hashtag we search. One is the number of the tweets containing the hashtag overtime. The rest one is Top 10 key opinion leaders of the hashtag on Twitter, ranked by their popularity scores*. (popularity scores = 2*number of retweets + number of likes + number of replies)

b) Google Data from Pytrends (API)

While web scraping Twitter data is a great method to understand how people perceive and feel about certain issues, it's limited and could be biased based on the demographics of Twitter users which trend towards younger people. Hence, we felt that also using Google search data would be a great complement to our Twitter data visualisations by providing a more complete picture of how certain issues are perceived and thought of by people. We utilise the Pytrends package in order to return Google search data that is similar to searching for search terms in https://trends.google.com/

We then used the data to create similar visualisations as we did with the Twitter Data. The first is a line graph that plots the search interest of a search term over a period of time. We use the same search term and same start/end dates as we do for our Twitter Historical data so that we are able to compare how search interest differs between Twitter and Google. We also chart the search interest of the Top 5 Related Topics to this search term over the same period of time. We also do a Word Cloud that is similar to the word cloud used for our Twitter data on our Related Queries data. Finally, we want to better understand the search interest of certain issues geographically so we chart the Top 20 countries based on search interest.