Project Report: The Climate Conversation on Social Media: Finding Trends in Tweets Related to Climate Change

Ashkan Aleshams, Ehsan Emadi, Jiajin Wu, Michael Galloro

Monday, December 14, 2020

Problem Description and Research Question

Many experts believe that global warming and climate change is happening. According to the information that we have collected from the United Nations website, the global average temperature raised by 0.85 centigrades from 1880 to 2012, and the sea's average level increased by 19 cm from 1901 to 2010. Every day we observe new climate events such as ocean acidification, droughts, floods, Greenland and West Antarctic ice melting, and many other undeniable climate phenomena that prove that climate change is real. Given this dramatic impact climate change will have on our future, it is inevitably a hot topic on social media as well.

Nowadays, most people are spending a vast amount of their time on social media. Based on the data that Clement has gathered in Statista, in 2019, on average, people spent 144 minutes per day on social media, and this number has been continuously increasing over the past few years (Clement, 2020). Since it is incredibly challenging to investigate all social media types and all kinds of comments about climate change, we decided only to check tweets and find trends among climate-change related tweets, which were found by scraping twitter data for climate change related hashtags.

While exploring this data set, we realized that our original research goals required a dramatic change. With the hashtags that the data set was built from, we had assumed there would be many climate-denying tweets with hashtags like "#climatechangeisfalse". Thus, our original thesis was: "What linguistic patterns can we observe in social media amongst climate change deniers?" Surprisingly however, we discovered that only about 1 in 50,000 tweets were identified with a 'climate-denying' hashtag, even after we expanded our pool of what could conceivably be considered a climate-denying hashtag (#MAGA for example). Given the sparsity of these climate-denying tweets, we knew it would be infeasible to explore the thesis we had originally planned. So by necessity, we generalized our analysis, and instead of focusing on climate-deniers, we instead set out to explore patterns in the entire body of climate change related tweets, and patterns in the people who are contributing to the climate change discussion on twitter.

Therefore, our revised thesis is as follows: **What observable trends are there amongst tweets related to climate change, and amongst the people who are contributing to the climate change conversation on twitter?**

Datasets

Tweet data file
Name: climate_id.txt
Source: Harvard Dataverse Organization Website -
//dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/5QCCUU
Description: A text file that contains twitter IDs related to climate change. Through a process called "hydration", the original tweet can be attained from their IDs. In order to work with the data and get access to the information we'll be using, we will convert it into JSON format through the Hydrator Platform which uses these ID's and Twitter API to get access to these tweets and put them into a dictionary. After converting, each tweet data is a dictionary with strings as keys and strings, int and dictionaries as values. These keys represent tweet data; such as date of tweet, name of user, text body, font, user description, etc.

Computational Overview

- **Major Computations:**
  - **Aggregating** dictionary of twitter info and **transforming** them into User and Twitter abstract class objects. In this process we also **filtered** out retweets to ensure each tweet is unique, along with a slew of unused tweet information, such as twitter ID.
  - **Aggregating** the following information from our list of Tweets objects: The number of times a hashtag was used, the number of times a word was said in every tweet, and in every user description, and the number of times a location was said in every user location. The aggregation algorithm for all of these were similar, using for loops and an accumulator. We also **transformed** the data to be all lower case and remove non-alphanumeric characters, to account for different formatting of the same words. The **filtering** process was more interesting and unique:
    - **Hashtags:** The data was collected using hashtags, so there were many very common ones such as #globalwarming. To make our results more useful, we used the `trim_max_values` function to filter out these very common hashtags.
    - **Description and Tweet Words:** We needed a powerful way to filter out common words such as "the". We discovered this is a common problem for algorithms, and the problem words are known as "stopwords". We implemented a large set of stopwords from XPO6, and customized it to our specific needs by including uninteresting words such as "global", "warming" and "world".
    - **Locations:** We created an **algorithm** in `clean_locations` which filters out common location words such as "city", and unifies spelling variations of common places, such as "ny" and "new york".
  - **Aggregating** sentiment scores for tweets (using VADER, described under new libraries), and then again aggregating these scores into useful subcategories in order to display them in a visually pleasing way.

- **Visualizing our Results:**
  We took our common hashtags, tweet words, user descriptions, and user locations, and made a word cloud with them. This easily showcases, in a visually pleasing way, a snap shot of the climate conversation on twitter, and some cool information about the users that are contributing to this conversation. We also made a pie chart, indicating how positive each tweet was (by VADER), from "extremely positive" (a score of 0.85 to the max score of 1) to "extremely negative" (a score of -0.85 to the minimum of -1). Finally we created a short contextual explanation of the visual data, and a longer explanation of VADER, including two fun examples of tweets that were at the extremes of the VADER analysis. All of the functions that implement these visualizations can be found in `visualizing_data.py`.

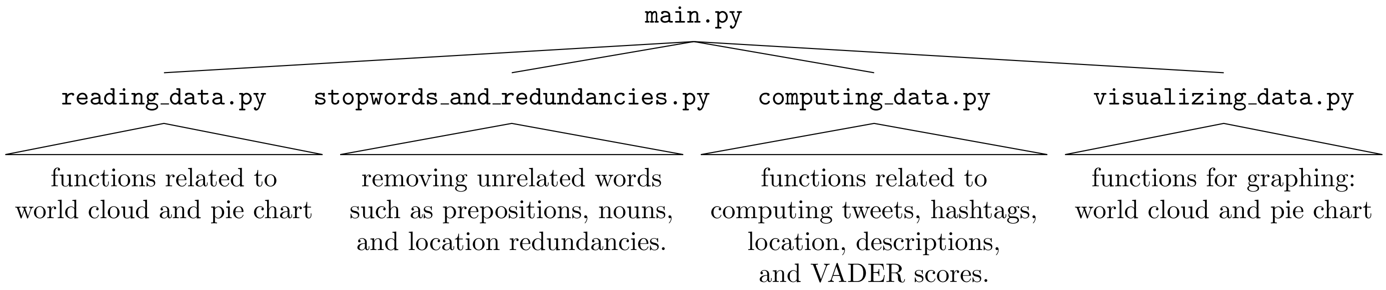
- **New Libraries:**
  - **VADER, Valence Aware Dictionary and sEntiment Reasoner**: This is a library we explored a simplistic version of in assignment 3. We decided to use the full library as a powerful tool for our project. Since it is specifically designed for social media text, we knew it would be a perfect fit. The library essentially runs a complex algorithm in order to assign a "sentiment" rating to text. So for example, "I love climate change" would have a positive score, and "climate change sucks" would have a negative score. This library helped us answer the question: "how positive are the tweets in the climate conversation?" We called the vader sentiment `polarity_scores` function in its `SentimentIntensityAnalyzer()` to analyze each tweet which its massive dictionary of word to sentiment value pairs. The VADER run time is very slow on a data set as large as ours, so as a compromise we ran it on subset of our tweets.
  - **Matplotlib and WordCloud:** WordCloud gave us a host of options for customizing our word clouds, including a custom shape drawn from any picture we choose, custom coloring using the `ImageColorGenerator()` class, , and custom "relative scaling" which makes the more common words either bigger or smaller in relation to the less common words. Matplotlib enabled us to plot the WordCloud data and visualize it in a window using the `plt.figure()` function, which includes a zoom option to see some of the less common words in our clouds. Matplotlib also allowed us to include text in our visualization, including titles with `plt.suptitle`, and our VADER examples. Finally, Matplotlib enabled us to make the pie chart, which took our aggregated VADER score groups and automatically proportioned them.
  - **numpy and PIL:** PIL (Python Imaging Library) allowed us to open our image files for the word clouds with `Image.open()`, and numpy's `np.array()` allowed us to transform the image into an array that the WordCloud could recognize and make use of for its shape.

Instructions for obtaining data sets and running the program

- Open **requirement.txt** and download all libraries/imports
- We strongly recommend using the data set we supplied with UTSend to avoid the long hydration process. You can email any of us if the UTSend expires.
- Otherwise, this is the link to the data set: *https : //dataverse.harvard.edu/file.xhtml?persistentId = doi :* 10.7910/*DVN*/5*QCCUU*/*QPYP*8*G*
- This is the hydrator: *https : //github.com/DocNow/hydrator/releases* Once the hydrator is installed, select the add tab, and locate the tweet ID file. Stop hydrating once you've gotten around 50,000 tweets.
- There's a total of 4 images on markus that the code requires, ensure these are unzipped and remain in their folder called **images**. Their file names should be: **earth.png**, **bird.png**, **speech.png**, and **brain.png**.
- **UTSend Info**:
  Claim ID: ndq2RQ83tmGzD8yg
  Claim Passcode: PFSDKjMp8NWppcTC
  The data file should be named **climate_id.jsonl**

What to expect from the report

This is a summary of the code structure in python:


main.py
reading_data.py
stopwords_and_redundancies.py
computing_data.py
visualizing_data.py
functions related to world cloud and pie chart
removing unrelated words such as prepositions, nouns, and location redundancies.
functions related to computing tweets, hashtags, location, descriptions, and VADER scores.
functions for graphing: world cloud and pie chart

After main has finished executing, read **figure 1** first, it provides more details and instruction for the visualizations. Some computers may not open all the figures simultaneously, in which case you must close a figure once you are done viewing it to see the next one. There is a total of six figures: a read me, followed by four word clouds, and finally a pie chart.

Changes

- We changed our entire thesis, for reasons described in our problem description.
- We changed our computation plan to include aggregating, transforming, and filtering user descriptions and user locations because we wanted to know what kind of people were tweeting about climate change for our new thesis.
- We changed our computation plan to a much more comprehensive filtering plan, for more useful results.
- We changed our presentation plan to include more word clouds for each category, to visualize the VADER date with a pie chart, and to include the VADER examples for more context.

Discussion

The hashtag, tweet word, and VADER visualizations are what helped us answer the first part of our research question: finding trends in the tweets themselves.

With the hashtag wordcloud We discovered that #auspol, which stands for australian politics, was a hot topic on twitter when this data was collected, in 2017-2019. Australians had a very strong presence in the climate conversation. #heatwaves and #wildfires trended as popular concerns. #renewables, #sustainability, and #cleanenergy trended as popular solutions. #parisagreement, #stopadani, and #trump trended as popular political topics related to climate change. For context, #stopadani is a movement in australia to stop the Adani coal mine from operating.

Next, the "tweet word" word cloud, a word cloud of the most common words in the actual tweets, had Ryan "Zinke", a US politician, as a popular word because he controversially blamed environmentalists for wildfires. "Blame" was also popular, which shows a pattern in the tone of the conversation.

The VADER visualization, despite being one of the more computationally complex aspects of our report, made for uninspiring results. We found that the climate change tweet sentiments had a relatively normal distribution, neither leaning positive nor negative. This surprised us as we had presumed it would trend negatively since it was such a heated topic.

The second part of our thesis was about trends in the *people* contributing to the conversations. The location and description word clouds helped us find these. These people described themselves as energetic, full of life, loving, writers, activists and interested in science, news and politics. The top locations were USA, Europe, and more surprisingly due to their lower populations, Canada, Australia, and Finland.

Optimization of run time was a big hurdle and important goal of ours. We initially used a library called NLTK for our stopwords, but found that it was too slow, and since we needed to check if each word of every tweet was a stopword, in 50,000 tweets, a slow stopword library was out of the question. Our solution was to take advantage of the constant search time of sets by implementing our own set of stopwords which we sourced online and then customized for our own needs. Our `trim_max_values()` function also required run time optimization. It's purpose was to take a dictionary with string to integer pairs, and remove the entries with the highest integers. We initially used a simple for loop and accumulator, but we discovered it was much faster if we first put the dictionary keys and values into a list, and then took advantage of list indexing to find the key associated with each max value. The results of our improvements were invaluable to being able to feasibly compute more tweets. If not for the slowness of converting a json file into a dictionary, we could have computed many more tweets.

Something seemingly very simple, showing a random tweet text on matplotlib, provided as another obstacle for us. We can not call it a limitation of matplotlib, as we are too inexperienced to know the full functionality of the library. Nonetheless, we struggled because of random "newline" prompts in tweets that affected matplotlib's formatting because it read and executed those same prompts. Oversized emojis also proved to be unpredictable. Our solution was to limit the tweets that could be shown.

Finally, a limitation we discovered in a data set with user created data such as this one is that it is very difficult to transform the data so that like terms are grouped together. For example, the location "new york city" was expressed as "ny", "new york", "new york, new york", and many other variations. Calculating precisely how many people lived in New York City was not an achievable task, we could only approximate. We used custom filters and transformation algorithms to help with this problem, but we could not hope to cover every corner case.

A natural next step in exploring our question further would be to improve our filtering and transforming algorithms so the results of our computations are more meaningful. For example, we could further customize our set of stopwords, or make our algorithm for grouping like terms that is more effective. Additionally, we could optimize the process of reading json files, or find another way to turn tweets into python code. Then, we could feasibly compute on a much larger data set. This would allow us to explore trends for more specific uses like our original thesis about people who deny climate change. Lastly, we want to explore the VADER findings even further. Specifically, we want to know why tweets on such a hot topic had a neutral sentiment. Is climate change not as much as emotionally charged as we had presumed? Is VADER not up to the task of deciding a human's sentiment through text? Or maybe, could we do a deep dive of the extreme of these sentiments, the extremely positive and negative tweets, and find some interesting climate change related patterns there?

References

Brahaj, A. (2019, April 29). List of English Stop Words. XPO6. http://xpo6.com/list-of-english-stop-words/.

Cdimascio. (n.d.). Cdimascio/py-readability-metrics. Retrieved November 04, 2020, from https://github.com/cdimascio/py-readability-metrics

Clement, J. (2020, February 26). Daily social media usage worldwide. Retrieved November 03, 2020, from https://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/

Climate Change – United Nations Sustainable Development. (n.d.). Retrieved November 05, 2020, from https://www.un.org/sustainabledevelopment/climate-change/

Cjhutto. (n.d.). Cjhutto/vaderSentiment. Retrieved November 04, 2020, from https://github.com/cjhutto/vaderSentiment

Vu, D. (Tutorial) Generate Word Clouds in Python. DataCamp Community. https://www.datacamp.com/community/tutorial
python.

Kasap, Mechmet. "Plotly (Scatter, Bar, Pie Chart) and Word Cloud," Retrieved October 15, 2018. https://www.kaggle.com/m
scatter-bar-pie-chart-and-word-cloud/notebook.

Littman, J., & Wrubel, L. (2019, May 20). Climate Change Tweets Ids. Retrieved November 04, 2020, from https://doi.org/10.7910/DVN/5QCCUU

Westhoek, H., Lesschen, J., Rood, T., Wagner, S., Marco, A., Murphy-Bokern, D., Oenema, O. (2014, March 26). Food choices, health and environment: Effects of cutting Europe's meat and dairy intake. Retrieved December 13, 2020, from https://www.sciencedirect.com/science/article/pii/S0959378014000338