ATinstagram

Team:
Nicholas Halstead, Steve Cho, Mason Barden, Tashi Jeshong, Zubin Joseph

Client:
Morva Saaty

Instructor:
Edward A. Fox

CS 4624 Multimedia, Hypertext, and Information Access
Department of Computer Science
Virginia Tech, Blacksburg VA 24061

May 9, 2022

Table of Contents

List of Figures

List of Tables

1.0 Abstract

Hiking the Appalachian Trail (AT) is an adventurous goal that many people around the world aspire to complete. Doing so requires meticulous planning and advice from those who have already hiked the Appalachian Trail. Social media can provide helpful information, describe unique experiences, and create a community focused on a variety of topics, such as the AT.

For this project, we wanted to discover if and how hikers use the social media platform, Instagram, to talk about Leave No Trace (LNT) principles on the Appalachian Trail. Leave No Trace principles refer to a set of guidelines that hikers should follow in order to promote conservation on trails. One principle relates to proper waste disposal. When hiking, you should be mindful of disposing of anything that you bring, whether that be food, trash, or hygiene products.

Using the description of the project outlined by our sponsors, a workflow to complete the project included: collecting relevant Instagram posts, performing sentiment analysis on these posts, and finally, creating a series of graphs that show the different connections between posts. We started by utilizing Python, JSON objects, and Selenium to gather all of the Instagram posts with specific hashtags, such as “#AppalachianTrail”, “LeaveNoTrace”, and “LNT”. Selenium is used for the API calls, which retrieve the many Instagram posts. Information about each post, such as its geographic location, caption, and hashtag, is extracted using JSON objects. It was then important to clean that data by eliminating posts that would not help with our analysis later on. Examples of this included posts not in English and posts that were clearly spam or advertisements.

The final two parts of the project include performing sentiment analysis on the collected posts and then visualizing the data in a variety of ways. For the sentiment analysis, we analyzed each caption of every post, and assigned it a score ranging from negative one to positive one. Negative one would represent a highly negative sentiment and positive one represents a highly positive sentiment. From there, we utilized the K-Means clustering algorithm to gather posts with similar hashtags. For the visualizations, we displayed what tags occur in the same post, connections between different hashtags, and the geolocations of the different posts. The deliverables of our project include the source code that is used to scrape the Instagram posts, perform sentiment analysis, and visualize the data, along with folders showing the results of our data collection. These results include the scraped Instagram posts, the sentiment analysis results, and the visualizations we created. These deliverables could help our client and those interested with research relating to Instagram, Leave No Trace principles, and the Appalachian Trail.

2.0 Introduction

2.1 Objective and Background

Instagram [1] is a social media platform used by millions of people who post on a diverse range of topics. Our sponsors are interested in posts related to the Appalachian Trail, specifically about Leave No Trace principles [2]. Our objective is to gather all of the relevant Instagram posts based on specific hashtags, perform sentiment analysis on every post with a caption, and then visualize the data we've collected throughout the semester. Once completed, we are to hand over all of the completed work to our sponsors, so they can further analyze the data and continue their research regarding Leave No Trace principles and the Appalachian Trail.

2.2 Deliverables

As previously stated, we are completing this project so our sponsors can use the data and visualizations we produced to further their research regarding the Appalachian Trail. To complete it, we created the following deliverables:

1. A .csv and a .txt file containing all of the collected and cleaned Instagram [1] posts relevant to Leave No Trace [2] principles.
2. Several .csv and .txt files containing all of the collected data, separated by individual hashtags.
3. Several visualizations that relate to the geolocation of different posts, the connection between different hashtags, and tags that occur within an individual post.
4. Our entire code base, in case the sponsor and her team need to make any updates or collect any data in the future.

2.3 Client and Team

Our main client and primary contact for this project is Morva Saaty. She is a Computer Science Ph.D. candidate at Virginia Tech. Other contacts for this project include: Scott McCrickard, Kris Wernstedt, Shalini Misra, and Jeff Marion. Scott McCrickard is a professor at Virginia Tech, within the Computer Science department. Kris Wernstedt and Shalini Misra are associated with Virginia Tech's School of Urban & International Affairs. Finally, Jeff Marion is associated with USGS & the Department of Forest Resources.

Our team members include: Steve Cho, Nicholas Halstead, Mason Barden, Tashi Jeshong, and Zubin Joseph. Every member of the team is a senior studying computer science, with interests in the Appalachian Trail and Instagram. Tashi is our team lead, and is focused on setting up meetings, ensuring the team is meeting its deadlines and goals, and helping with testing and implementation. Nicholas and Steve are the heads of implementation, and have spearheaded a majority of the development work. Mason was focused on helping with implementation as well as documentation of the project. Finally, Zubin is tasked with leading the documentation aspect of the project, and helps with testing.

3.0 Requirements

Our project had a variety of requirements in order to complete the project:

- Collecting data from Instagram concerning sustainable practices and AT hikers' experiences in the timeframe 2018-2021
    - Using different hashtags (e.g., #atclass2021, #leavenotrace, #atnobo2020, etc.)

- Performing Sentiment Analysis on collected data to extract barriers and challenges hikers encountered on the trail and benefits they seek to get through hiking.
    - Use continuous scores (positive, slightly positive, neutral, slightly negative, negative).
    - Extract captions and related words that resulted in categorizing the positive and negative posts.

- Visualizing the connection between collected hashtags in a meaningful way using a network graph.
    - Representing which hashtags are coming together in the posts and how frequently they have been used by users. For doing this, students need to use K-Means clustering and proper visualization techniques.

- Collecting geolocation data from users' posts on Instagram and visualizing these data based on different years to better understand AT hikers' movements over various times.

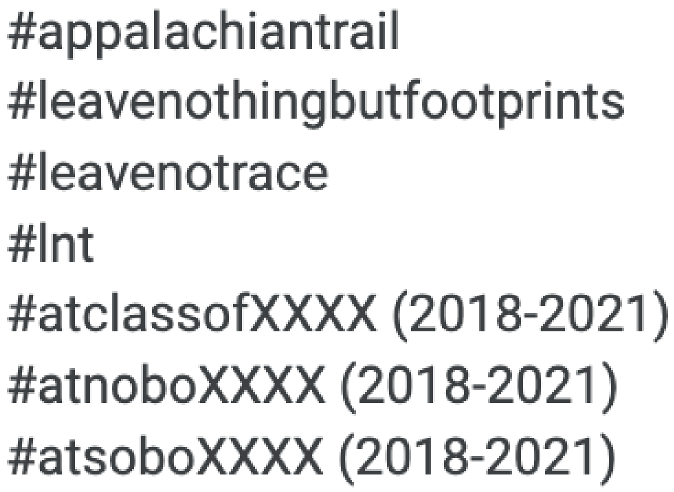
4.0 Design & Implementation

4.1 Initial Data Collection

The first step in the project was to figure out how to obtain data from Instagram [1]. Python [3] libraries that involved scraping web data failed to produce reliable results since Instagram requires login credentials to be provided after less than a dozen requests from a particular IP address. To remedy this problem, our team used a library, Selenium [4], that would create an instance of Google Chrome [5] that could be controlled by code. This allowed for posts data to be scraped using an old API in-browser since login information could be passed through the browser as cookies.

4.2 Automated Data Collection

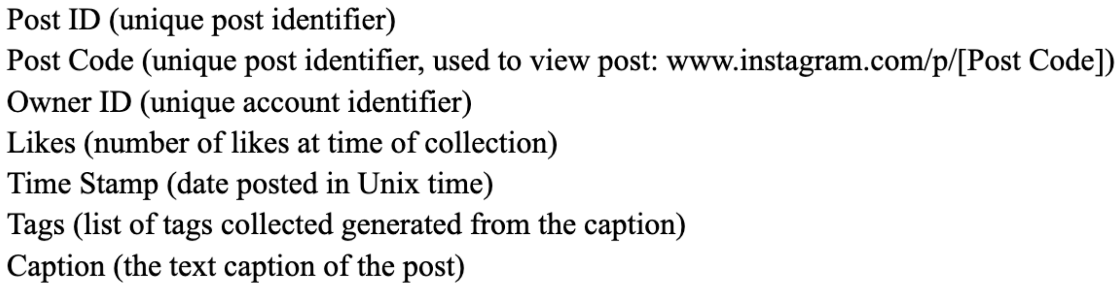
The setup of Selenium [4] and more direction from the sponsors allowed for large-scale data collection to begin. The sponsors gave us a list of tags to collect posts from in addition to the relevant tags determined by the team. The final list of tags collected is shown in Figure 1:



#appalachiantrail
#leavenothingbutfootprints
#leavenotrace
#lnt
#atclassofXXXX (2018-2021)
#atnoboXXXX (2018-2021)
#atsoboXXXX (2018-2021)

Figure 1: List of Hashtags.

The code written was designed to search for the most recent posts under a certain tag and record the last "end cursor" [6] that was accessed for future data collecting. This end cursor is a value that acts as a key to obtain the next page of posts. The post data was stored in a labeled, unique file (guaranteed by using a Unix timestamp [7]), and the metadata for the query was stored in a matching file under a different folder. In total, roughly 150,000 Instagram posts were collected. The specific attributes collected are detailed in Figure 2:



Post ID (unique post identifier)
Post Code (unique post identifier, used to view post: www.instagram.com/p/[Post Code])
Owner ID (unique account identifier)
Likes (number of likes at time of collection)
Time Stamp (date posted in Unix time)
Tags (list of tags collected generated from the caption)
Caption (the text caption of the post)

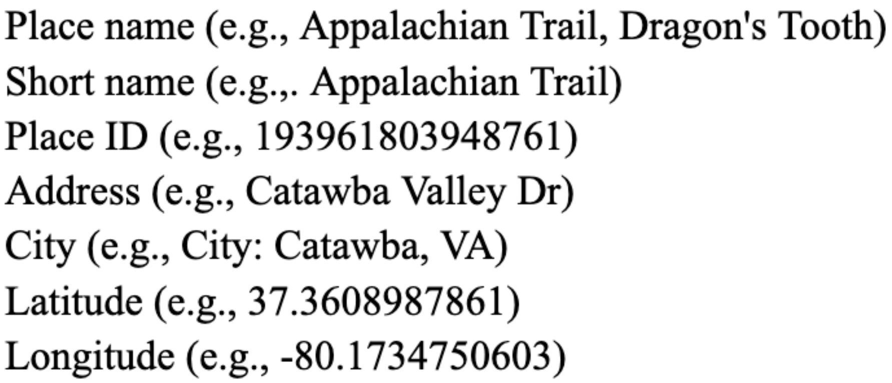
Figure 2: Collected Attributes from Instagram Posts.

4.3 Data Cleaning

The sponsors indicated that not all of the data we collected was of use to them, namely posts written in a language other than English. Additionally, our team found that some of the posts collected under tag X did not include tag X in its caption. Rather, the API functioned such that posts containing tag X in the comments would still appear in the API query. These posts were potentially not related to tag X in the first place and were cleared to be discarded by the sponsors. The Python library langdetect [8] was used to determine which posts were confidently not in English (erring on the side of leaving false negatives alone). This brought our dataset down to roughly 120,000 Instagram [1] posts.

4.4 GeoData Collection

The automated data collection process was able to get most information from each post, but one crucial piece of information was not included in this API request: geolocation data. This had to be obtained by requesting information on a specific post using its post code. While the original data only took roughly 3,000 requests to build up (about 50 posts per request), the geolocation data would take 120,000 requests (one per post). This had to be spread out over multiple days and Instagram [1] accounts to avoid tripping the system into thinking we were a denial of service attack [9]. Roughly 50,000 posts had geolocation data available, and this was collected into a separate file. This file is found in the geodata folder and represents all the geodata our team collected. There are some challenges beyond our control such as posts that are no longer available since time of initial collection and posts that do not have any geolocation data attached to them. The specific attributes collected from this geolocation data collection process are detailed in Figure 3:

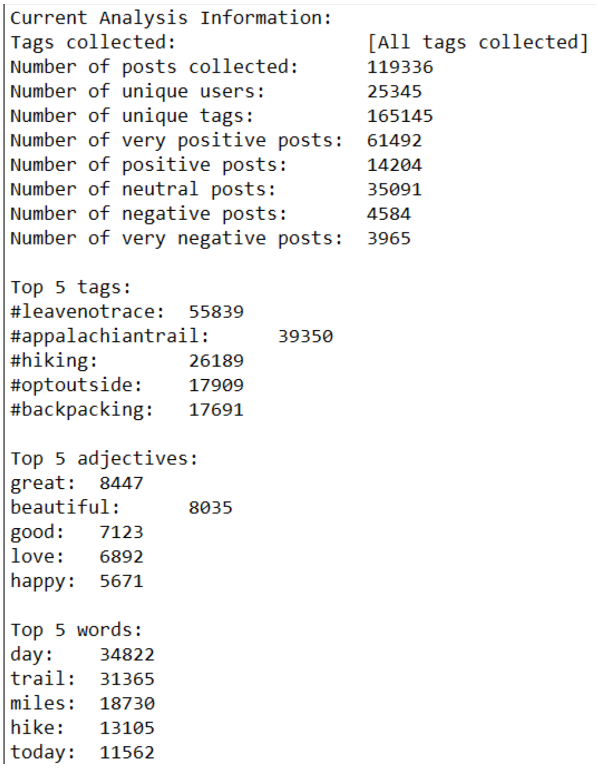


Place name (e.g., Appalachian Trail, Dragon's Tooth)
Short name (e.g.,. Appalachian Trail)
Place ID (e.g., 193961803948761)
Address (e.g., Catawba Valley Dr)
City (e.g., City: Catawba, VA)
Latitude (e.g., 37.3608987861)
Longitude (e.g., -80.1734750603)

Figure 3: Attributes from Geolocation data Collection.

4.5 Initial Data Analysis

The initial analysis of the data was crude and aided mainly to understand the data we collected. Preliminary information was collected, such as the number of unique users, posts, and tags. The top tags and words used in the dataset were also collected, which would be used later for other analyses. The initial data analysis as a summary of the dataset used is shown below:



```
Current Analysis Information:
Tags collected:                 [All tags collected]
Number of posts collected:      119336
Number of unique users:         25345
Number of unique tags:          165145
Number of very positive posts:  61492
Number of positive posts:       14204
Number of neutral posts:        35091
Number of negative posts:       4584
Number of very negative posts:  3965

Top 5 tags:
#leavenotrace:  55839
#appalachiantrail:      39350
#hiking:        26189
#optoutside:    17909
#backpacking:   17691

Top 5 adjectives:
great:  8447
beautiful:      8035
good:   7123
love:   6892
happy:  5671

Top 5 words:
day:    34822
trail:  31365
miles:  18730
hike:   13105
today:  11562
```

Figure 4: Preliminary Information Collected

4.6 K-Means Data Analysis

The K-Means analysis was a means to organize and analyze the data. Unfortunately, the sponsors of the project were more interested in other means of analyzing the data collected than this particular analysis and have asked us over our weekly meetings to focus on other work over this. As a consequence, the K-Means analysis framework has been constructed, but no concrete conclusions have been derived within the scope of this report. The limited work done with K-Means will be outlined here, but further analysis is left undone as future work for this project.

The first step in K-Means analysis is identifying a bag of words. Accordingly, we identified a collection of relevant words that could separate the posts into different groups based on their meaning and relevance to the project. Since the sponsors were initially curious about posts that best promoted Leave No Trace (LNT) sentiments, we added several words to our bag related to such feelings. For example, words such as “trace”, “minimize”, “respect”, “trash”, and “protect” were added to the bag of words. We sought to split the data into two groups: posts that supported LNT principles and those that did not. After initial testing (to be expanded upon in Section 5) led us to increase the number of clusters to encapsulate different clusters better, we found that 5 clusters split the data better than 2. The determination of this was a function that calculated the top words (and their frequencies) of each cluster. The top three words for each cluster with their frequency and how many posts were in that cluster can be found in Table 1.

Cluster	Words + Frequencies	Number of posts
1	wildlife: 0.007244 respect: 0.004088 trace: 0.001884	103,074
2	trash: 0.4886 visit: 0.3632 pick: 0.2281	5,404
3	better: 0.6908 plan: 0.3369 trash: 0.04710	3,991
4	protect: 1.0 wildlife: 0.1180 trash: 0.1008	780
5	leave: 0.9987 trace: 0.5297 trash: 0.1167	6087

Table 1: Initial Results from K-Means Analysis with K=5

4.7 Sentiment Analysis

One of the important features that the project required was to determine the sentiment of the Instagram posts. We used an open-source program VADER (Valence Aware Dictionary and Sentiment Reasoner) [10], which is based on rules and lexicons. The program is specifically designed to perform analysis on posts on social media.

4.7.1 Scoring

The sentiment scoring is based on identifying words in the sentence and their mean valence scores [11]. The 'vader_lexicon.txt' file contains over 7500 lexemes with mean valence score, standard deviation, and raw rating in order. With a given input sentence, the program identifies words that are in the lexicon list [12]. If there are lexemes that are written in all uppercase letters, the program applies scaling values to increase their intensity.

The negation words [13] are also accounted for when applying scaling values. If the words in the sentence are part of the negation words list, the program looks for adjacent words and applies negative scaling values to those words. In cases where there is more than one negative word, the program looks for a conjunction that continues the negativity of the words.

After gathering sentiment scores of the lexemes, the program calculates the positive, negative, and neutral scores by counting the positive, negative, and neutral summation. The total score is calculated by normalizing the summation of all sentiment scores.

4.7.2 Total scoring

The compound score is calculated by adding all valence scores of each lexeme, adjusted according to the rules. Then the score was normalized between negative one and positive one. The standardized thresholds for sentiment classification were set to:

- Positive sentiment: **total score** >= 0.5
- Slightly positive sentiment: 0.25 =< **total score** < 0.5
- Neutral sentiment: -0.25 < **total score** < 0.25
- Slightly negative sentiment: -0.5 < **total score** =< -0.25
- Negative sentiment: -0.5 >= **total score**

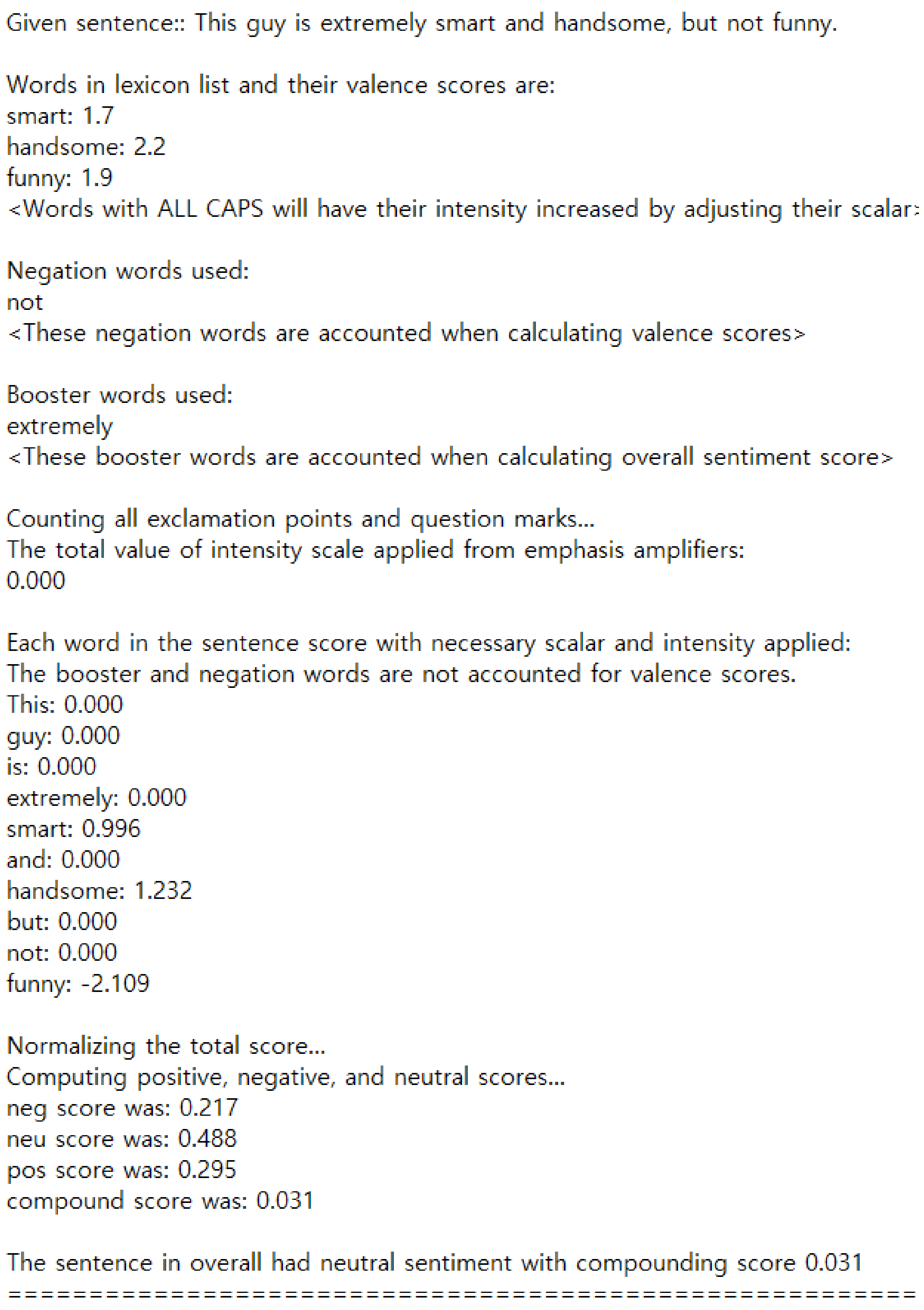
The scoring boundary and threshold values are different from the ones VADER [10] suggested, because we had to divide into five instead of three categories. Slightly positive and negative sentiments were added later on after feedback from our sponsors. This change in threshold values is explained in Section 5.3.2 in greater detail.

4.7.3 Booster Words and Emphasis Amplifiers

Booster words [14] are identified as a list of adverbs that would increase the intensity of overall sentiment scores in a drastic manner. Since the booster words are not in the lexicon list, they will give zero valence score on the result. The list contains the most used words in social media posts. Each adverb in the list is categorized to either increase or decrease the sentiment score. In addition, emphasis amplifiers are exclamation points and question marks that are used to alter the overall sentiment of the sentence. The program counts all exclamation points and question marks and scales the score with a multiplier, adjusting the intensity rating.

4.7.4 Modification

Original output from VADER [10] only showed sentiment scores consisting of positive, negative, and neutral scores of the text with compounding scores. Because of this, even though the example texts that we tested met the expectation in terms of their scores, we were not able to provide detailed results of the analysis. Our goal of the sentiment analysis was to figure out not only the sentiment score, but also which word in the sentence was responsible for the scoring.



```
Given sentence:: This guy is extremely smart and handsome, but not funny.

Words in lexicon list and their valence scores are:
smart: 1.7
handsome: 2.2
funny: 1.9
<Words with ALL CAPS will have their intensity increased by adjusting their scalar>

Negation words used:
not
<These negation words are accounted when calculating valence scores>

Booster words used:
extremely
<These booster words are accounted when calculating overall sentiment score>

Counting all exclamation points and question marks...
The total value of intensity scale applied from emphasis amplifiers:
0.000

Each word in the sentence score with necessary scalar and intensity applied:
The booster and negation words are not accounted for valence scores.
This: 0.000
guy: 0.000
is: 0.000
extremely: 0.000
smart: 0.996
and: 0.000
handsome: 1.232
but: 0.000
not: 0.000
funny: -2.109

Normalizing the total score...
Computing positive, negative, and neutral scores...
neg score was: 0.217
neu score was: 0.488
pos score was: 0.295
compound score was: 0.031

The sentence in overall had neutral sentiment with compounding score 0.031
==========================================================
```

Figure 5: Sentiment Analysis on a Sentence.

Figure 5 depicts an example of how VADER performs sentiment analysis on a sample sentence. The modified analysis result consists of identifying lexicons and their valence scores from a given input sentence, negation words in the "NEGATE" list, booster adverbs in "BOOSTER_DICT", exclamation points, question marks, each lexeme in the sentence with changed valence scores, and sentiment scores with the compounded score.

4.7.5 Statistical Results

We wanted to determine the number of each sentiment post, depending on the inclusion of specific hashtags. The tables below show the numerical values of each sentiment post which includes the 'Leave No Trace' hashtag and specific hashtags that we discussed with our client. These hashtags are: #leavenothingbutfootprints, #atsoboXXXX, #atnoboXXXX, atclassofXXXX, and #appalachiantrail.

	#lnt or **#leavenotrace**	**#leavenothingbutfootprints**
Positive Posts	27016 (**53.2%**)	691 (**50.8%**)
Slightly Positive Posts	6503 (**12.8%**)	137 (**10.0%**)
Neutral Posts	14209 (**28.0%**)	411 (**30.2%**)
Slightly Negative Posts	1719 (**3.4%**)	53 (**3.9%**)
Negative Posts	1386 (**2.7%**)	68 (**5.1%**)
Total Posts Collected	50833	1360

Table 2: Number of posts with #lnt and #leavenothingbutfootprints.

	#atsobo2018	**#atsobo2019**	**#atsobo2020**	**#atsobo2021**
Positive Posts	1 (**50.0%**)	0	1 (**100.0%**)	48 (**42.9%**)
Slightly Positive Posts	0	0	0	15 (**13.4%**)
Neutral Posts	1 (**50.0%**)	0	0	35 (**31.2%**)
Slightly Negative Posts	0	0	0	8 (**7.1%**)
Negative Posts	0	0	0	6 (**5.4%**)
Total Posts Collected	2	0	1	112

Table 2: Number of posts with #lnt and #atsoboXXXX.

	#atnobo2018	**#atnobo2019**	**#atnobo2020**	**#atnobo2021**
Positive Posts	22 (**81.4%**)	0	0	73 (**26.1%**)
Slightly Positive Posts	2 (**7.5%**)	0	0	15 (**5.4%**)
Neutral Posts	1 (**3.7%**)	2 (**100.0%**)	0	185 (**66.1%**)
Slightly Negative Posts	1 (**3.7%**)	0	0	1 (**0.4%**)
Negative Posts	1 (**3.7%**)	0	0	6 (**2.0%**)
Total Posts Collected	27	2	0	280

Table 3: Number of posts with #lnt and #atnoboXXXX.

	#atclassof2018	**#atclassof2019**	**#atclassof2020**	**#atclassof2021**
Positive Posts	6 (**75.0%**)	21 (**41.2%**)	4 (**36.4%**)	89 (**55.3%**)
Slightly Positive Posts	0	5 (**9.8%**)	3 (**27.3%**)	24 (**14.9%**)
Neutral Posts	2 (**25.0%**)	22 (**43.1%**)	3 (**27.3%**)	43 (**26.8%**)
Slightly Negative Posts	0	3 (**5.9%**)	0	3 (**1.8%**)
Negative Posts	0	0	1 (**9.0%**)	2 (**1.2%**)
Total Posts Collected	8	51	11	161

Table 4: Number of posts with #lnt and #atclassofXXXX.

	#appalachiantrail
Positive Posts	796 (**47.9%**)
Slightly Positive Posts	163 (**9.8%**)
Neutral Posts	594 (**35.8%**)
Slightly Negative Posts	63 (**3.8%**)
Negative Posts	45 (**2.7%**)
Total Posts Collected	1661

Table 5: Number of posts with #lnt and #appalachiantrail.

The general trend of these searches was that positive and slightly positive posts had the most number of posts for each hashtag search, except for the combined search of the 'Leave No Trace' hashtag and #atclassof2019. Amongst all the posts, negative and slightly negative posts had the lowest number of posts. The results also showed that the hashtags, #leavenothingbutfootprints and #appalachiantrail, were commonly used with the 'Leave No Trace' hashtag in Instagram [1] posts.

4.8 Data Visualization

Visualizations of the data have been created based on both the geolocation data as well as the tags collected from the captions of the posts.

4.8.1 GeoData Visualization

After completing the process of collecting geolocation data, we retrieved about 50,000 posts with their latitudes and longitudes. We mapped this data in Figure 6, which displays all the Instagram [1] posts on a map of the world. However, our focus being the Appalachian Trail, Figure 7 zooms in on the Appalachian Trail to illustrate how a large amount of data falls somewhere along the trail. Points that fall outside of the Appalachian Trail can be easily explained, as our data collection process focused on tags beyond just the Appalachian Trail, such as the tags "#LeaveNoTrace" and "#LeaveNothingButFootPrints".

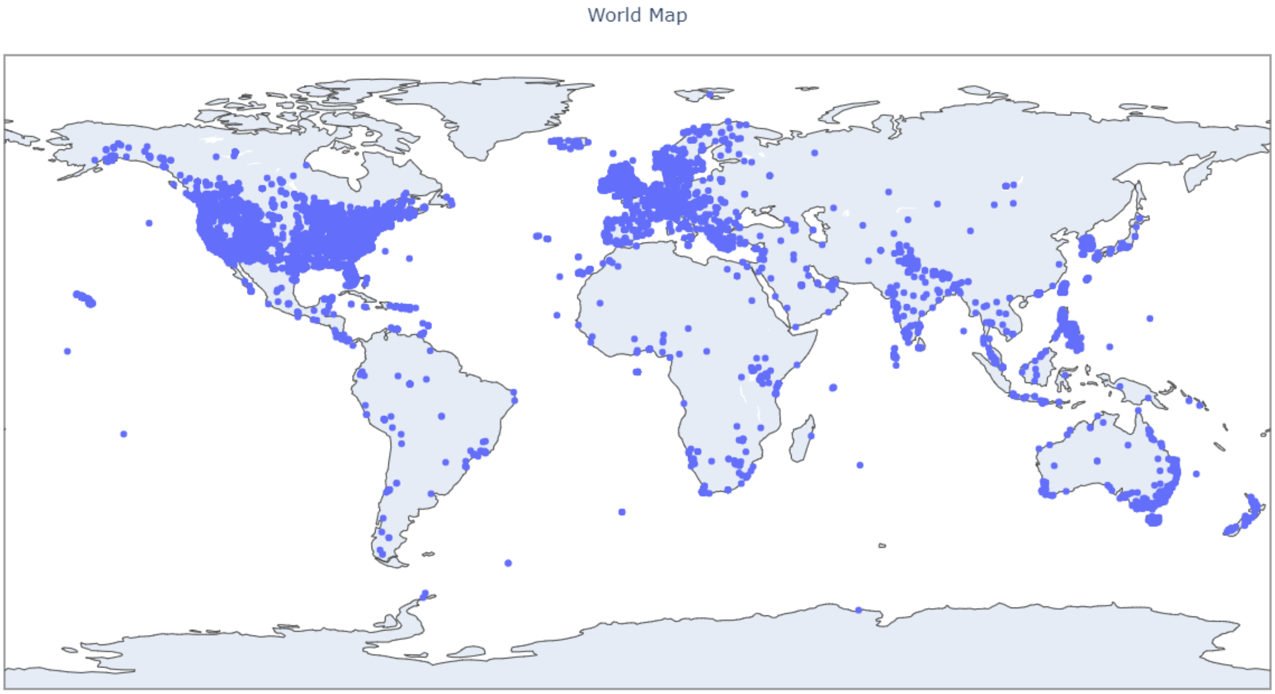


Figure 6: World Map with Post Locations.

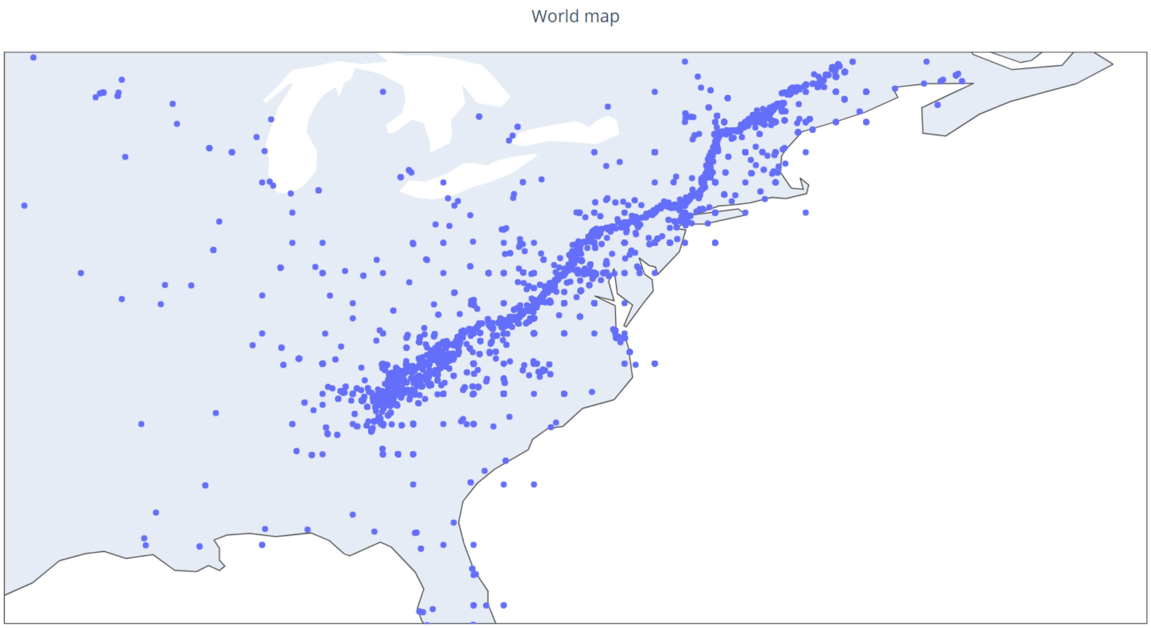


Figure 7: World Map of Posts, Focused on the Appalachian Trail.

4.8.2 Tags Graph Visualization

Another visualization performed was one that compared the relationship that certain tags had with each other. Figure 8 illustrates the top 100 most popular connections of tags, i.e., which tags were used most frequently with each other. This visualization is a dynamic one (the nodes can be moved around to view the text more clearly), but it has been condensed into a simple figure for this paper. To view the dynamic versions, open up the .html files located in the results folder and simply click and drag on any nodes you choose. Figure 9 illustrates the same as Figure 8, but now shows the top 150 most popular connections. The number of connections chosen to visualize was based on visibility and interpretability, as too many connections made the visualization too cluttered to understand.

The final visualization in Figure 10 first filters the most popular connections by those that include one of the tags we collected, and then displays the top 150 of those connections. This was done to highlight how tags such as #LeaveNoTrace, #AppalachianTrail, and #LeaveNothingButFootprints relate to both each other and other tags.

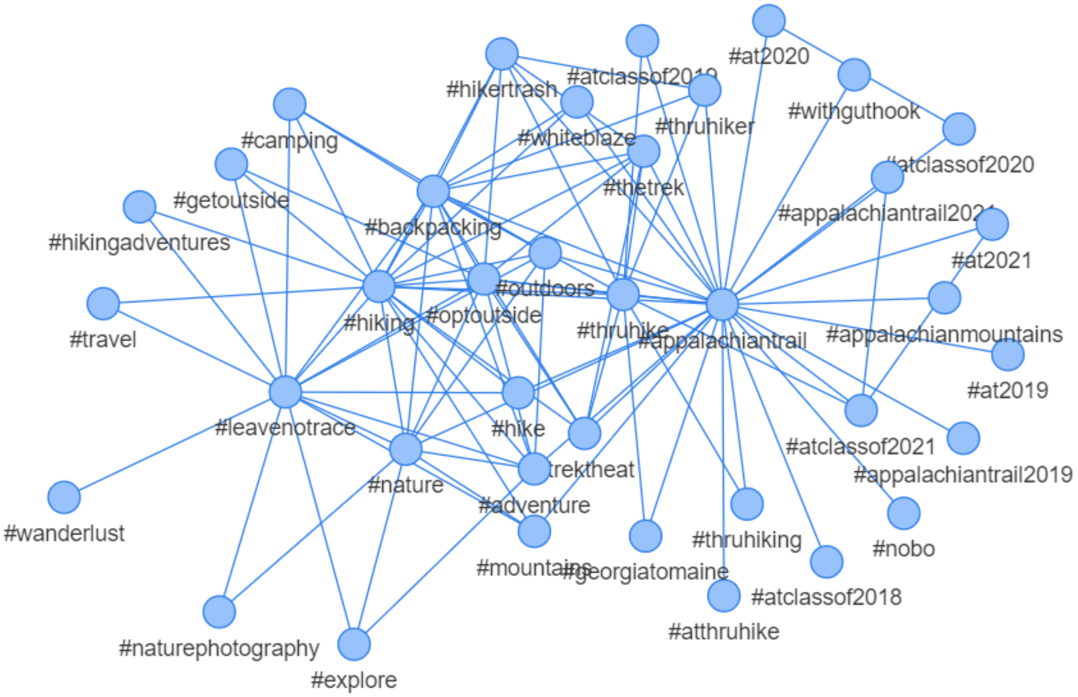


Figure 8: Top 100 related tags.

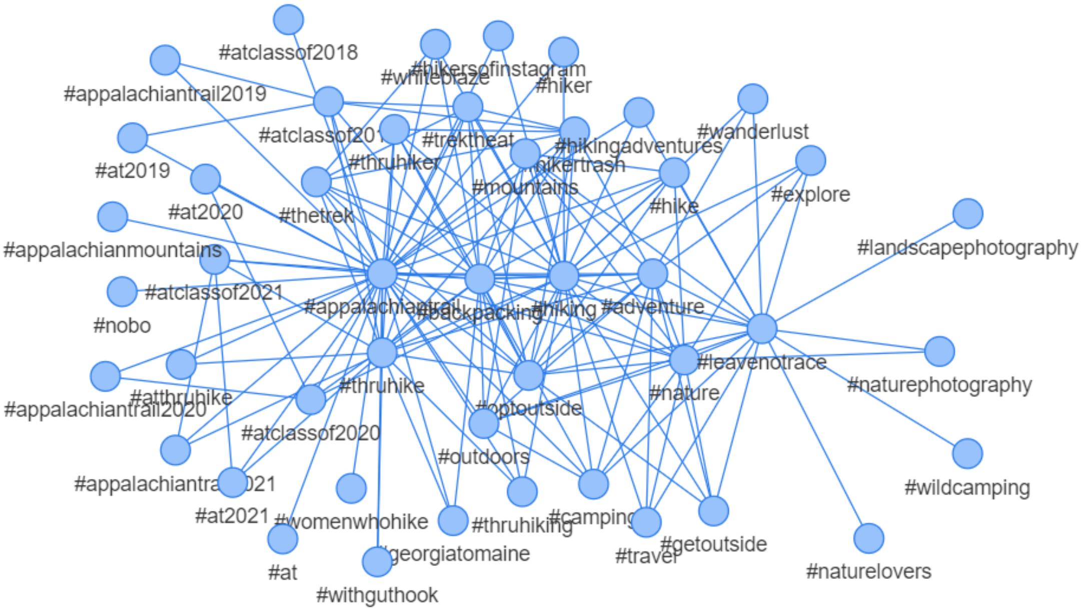


Figure 9: Top 150 related tags.

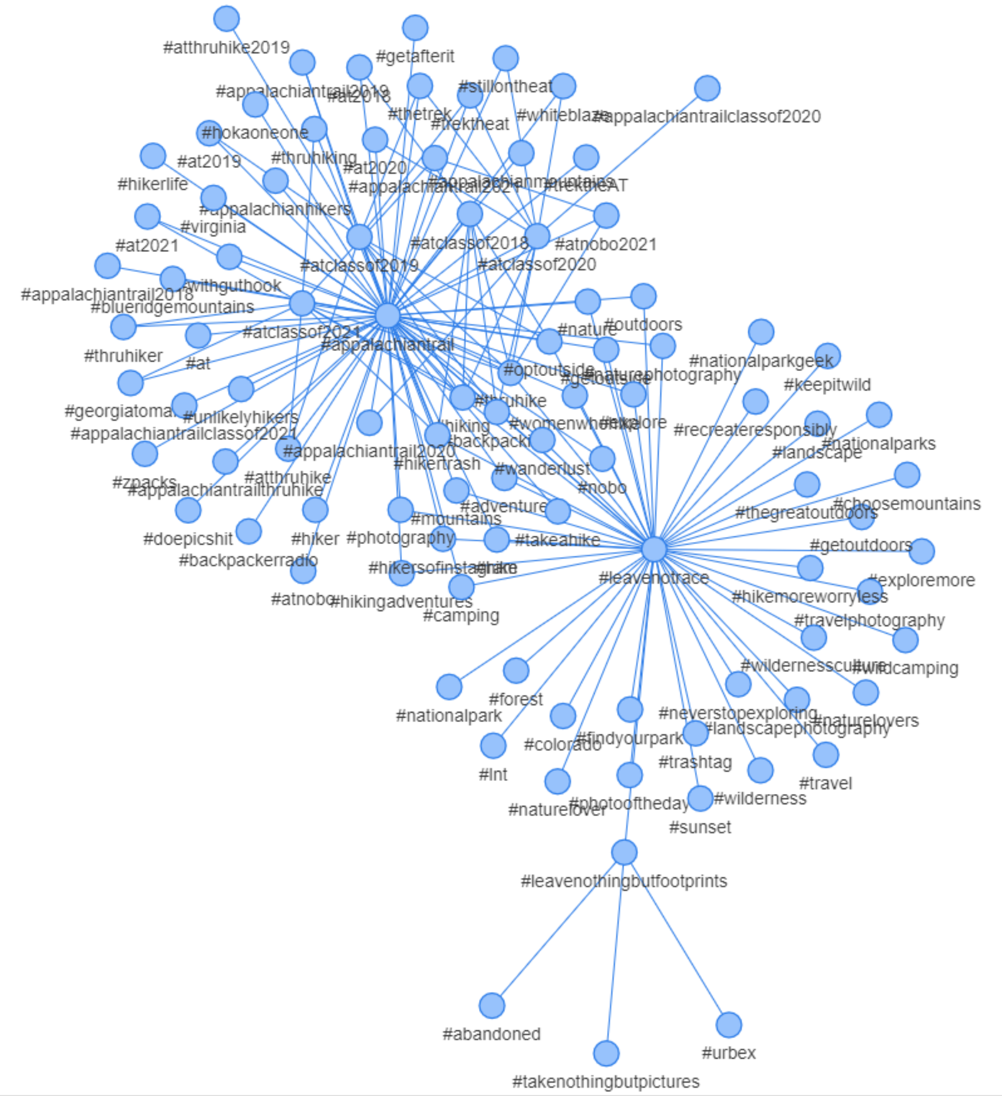


Figure 10: Top 150 related tags (focus on AT+LNT).

5.0 Testing

5.1 Post Collection Testing

Collecting the posts required a good amount of testing and evaluation. During the testing phase, it was discovered that the API would only allow for roughly a dozen posts without credentials. Selenium was used as a means to provide credentials through a program-controlled browser. Then, it was discovered that the credentials provided could not be shared; otherwise, Instagram would label the account as a proxy server and shut down or suspend the account. After creating a new account, we found that too many requests within a set time period (a thousand or so in twenty minutes) would make Instagram label the account used as spamming the API and would suspend API access for a couple of days. Data collected from multiple runs was then combined using a dictionary, using the post ID as a key, to ensure no duplicates.

5.2 Geolocation Testing

How we went about Geolocation testing is that we ran two different API requests to collect information about posts. The first API requests are discussed in Section 5.1. They were in the Post Collection Testing phase which collects general information about the post such as post ID, caption information, post code, etc. Starting with the first API request, we collected 120,000 total posts. However, in this first API request, Instagram did not provide information about the geolocation information on each post. Thus, to combat this problem, we wrote code to create a second API request that would specifically collect the geolocation information of each post. This second API requested geolocation information on each individual post from the 120,000 total posts collected from the first API request.

The second API requests for each post and then collects information such as the latitude, longitude, and city name. For an API request session, we had to create fake Instagram [1] accounts, and each account would call a maximum of 1,500 posts until Instagram times that account out and blocks the account for 24 hours. To solve this issue, we had to create numerous fake Instagram accounts.

During each step of processing 1,500 posts, the geolocation tagged post must be checked to ensure that the values are floats rather than ints, empty geolocation data must be handled, and a placeholder for the next iteration must be placed. We created code to remove duplication to prevent calling the same post twice.

As a result, we were able to acquire 50,000 geolocation tagged posts after parsing through 120,000 of the total collected posts.

5.3 Analysis Testing

Analysis testing is divided into two parts, K-Means analysis, and Sentiment analysis.

5.3.1 K-Means Analysis Testing

The K-Means algorithm was tested by running it with all of our data. Testing which bags of words and number of clusters provide the best results is something left as future work, but the testing done by this project will be detailed here. Firstly, creating a “bag of words” with more than a few hundred entities slowed down the process greatly. This is because each word is converted into an element of an array (the length of the bag). This array is then instantiated by checking each “bagged word” against each word in each caption in each post. Secondly, using word fragments, partial words or stems of words used to represent several forms of the same word (“leav” in place of “leave” and “leaving”) is ideal for the analysis, but it makes the process of vectorizing posts by the bag of words a longer process. Third, a small custom bag of words is more efficient than using frequent words from the dataset even if stopwords (such as “the”, “and”, “or”, etc. are omitted). This is due to large overlap in words with meaning across posts (words such as “hike”, “trail”, and “day”). This bag can be found at the top of the file /src/BagOfWords.py in a variable called SPARSEBAG. Fourth, it is better to err on the side of having too many clusters rather than too few since posts of a unique kind or group could be relatively few in number (say only a few thousand relevant entries), and the classifier with a K=2 would find it less costly by its Squared Sum Error (SSE*) error function to ignore that group and place two centers where the majority of the data lies. This means that the algorithm could ignore those posts on the edge of our high-dimensional space by placing centers of clusters (when $k < 4$) close to each other and thus providing inconclusive results.

*The SSE is the sum of squared distances each post is from its cluster. This is a common metric used to determine both stopping conditions (such as when the decrease of SSE slows down) as well as fitness for a particular run of K-Means clustering.

5.3.2 Sentiment Analysis Testing

When we initially tested sentiment scores for the collected description of Instagram [1] posts, we detected that most of the sentiment scores were toward the extremes. The below standard shows the threshold values of what VADER [10] suggested when testing sentiment scores:

- Positive sentiment: **total score** >= 0.05
- Slightly positive sentiment: 0.025 =< **total score** < 0.05
- Neutral sentiment: -0.025 < **total score** < 0.025
- Slightly negative sentiment: -0.05 < **total score** =< -0.025
- Negative sentiment: -0.05 >= **total score**

Since the standard that VADER suggested had only three categories, we added slightly positive and slightly negative sentiment threshold values when testing.

	#leavenothingbutfootprints	**#appalachiantrail**
Positive Posts	878 (**64.6%**)	1017 (**61.2%**)
Slightly Positive Posts	3 (**0.4%**)	7 (**0.4%**)
Neutral Posts	311 (**22.9%**)	486 (**29.3%**)
Slightly Negative Posts	12 (**0.9%**)	8 (**0.5%**)
Negative Posts	156 (**11.2%**)	143 (**8.6%**)
Total Posts Collected	1360	1661

Table 6: Number of posts collected #leavenothingbutfootprint and #appalachiantrail with original threshold value suggested by VADER [10].

Table 6 shows the number of each sentiment post collected with the standard that VADER suggested. The hashtags used for this testing are identical to what we tested for our threshold values. Since other search hashtags with years (e.g., #atnoboXXXX, #atsoboXXXX, and #atclassofXXXX) did not have a sufficient number of posts like #leavenothingbutfootprints and #appalachiantrail, their results are not included in Table 6.

There was a significant difference between the number of positive and slightly positive posts, as well as negative and slight negative posts. The occurrence of this result was mainly because most of the sentiment scores we got were larger or smaller than the suggested threshold values from VADER. A possible explanation would be a number of sentences given as an input. Since the description of an Instagram post usually has more than a single sentence, there are more possible scoring factors to be calculated in terms of lexemes, negation words, booster words, and emphasis amplifiers. So, having multiple sentences as input would have a higher sentiment score than having a single sentence as an input. Therefore, we thought that the threshold values that VADER suggested were not appropriate for our analysis. Furthermore, the suggested threshold values were not absolute, but can be changed according to a situation. We adjusted the threshold values by 10 times the originally suggested threshold value from VADER.

5.4 Visualization Testing

While not much testing was done to visualize the maps, there were a number of insights gleaned from experimenting with the tag graph visualizations (these experiments can all be viewed as .html files among the files presented to the client in the results folder of the provided ZIP file). Firstly, the number of connections should not exceed 200 unless one is prepared for a mostly unreadable graph. Secondly, trimming the graphs to only include connections involving one of the hashtags our project focused on has both costs and benefits. One cost is obviously lost data on how other tags relate to each other, but the benefit is that many more tags overall and their connections to the hashtags collected are visible. See the visualization below to illustrate the challenge of showing too many connections in one visualization.

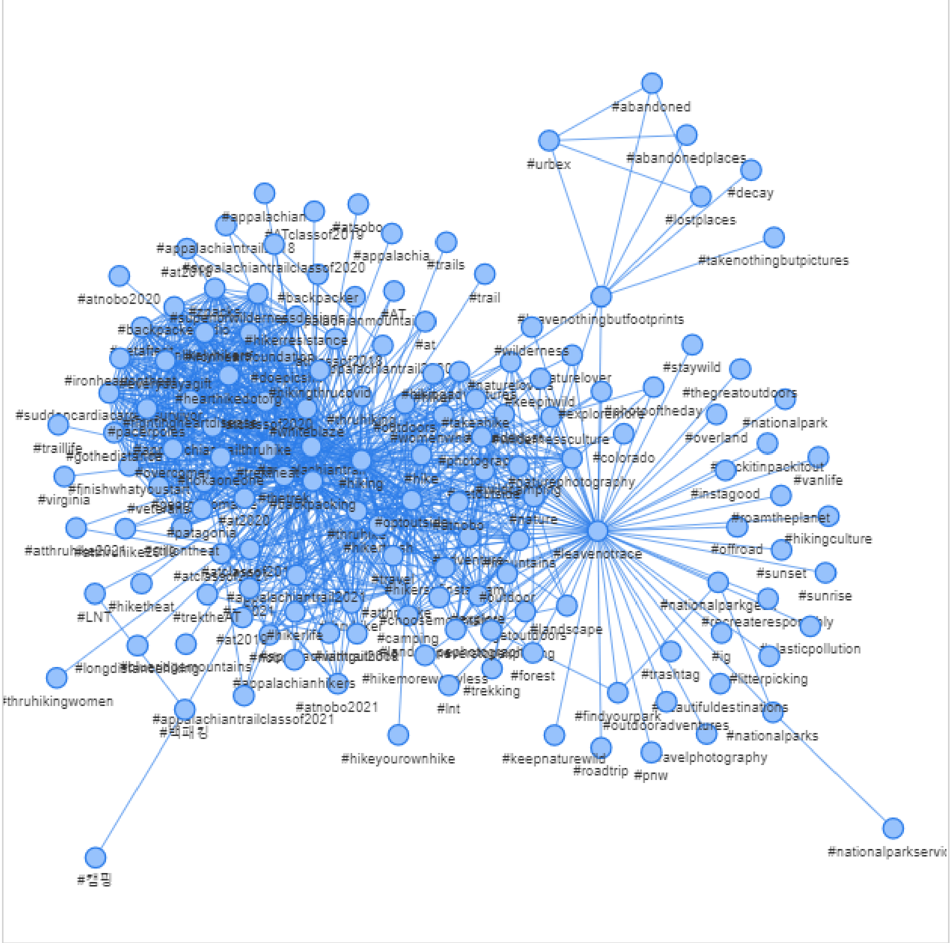


Figure 11: Top 1000 related tags (from all data).

6.0 Developer's Manual

6.1 Inventory of How Files are Structured:

- Auth
  - Credentials.txt - This is where you enter the username and password of the Instagram account you will use.

- Data - all of these are pre-existing data files that have been collected for reference.
  - Appalachiantrail
  - atclassofXXXX
  - atnoboXXXX
  - atsoboXXXX
  - leavenothingbutfootprints
  - Leavenotrace
  - LNT
  - CleanedCommonWords.txt
  - Negative.txt
  - Positive.txt
  - Sorted_adjectives.txt

- Results
  - Results_1646079898.zip - Example results information

- Src
  - Analyze.py - Analyze the data that has been collected
  - Analyze_OLD.py - Previous version for comparison.
  - BagOfWords.py - This creates a bag of words from the collected data.
  - CollectData.py - This collects the raw data.
  - CollectGeoData.py - This is the function that collects geographic data from the Instagram posts.
  - Utilities.py - This has helper functions that are used in the other scripts.
- Readme.md - Instructions for running the programs.
- Chromedriver.exe - The driver for Chrome version 97.

6.2 Dependencies

- Chromedriver.exe - Allows interaction between the code and Google Chrome.
- Python 3.8 - This is the Python version that our code base runs on.
- Selenium - This helps us with automation and scraping of Instagram posts.
- Langdetect - This allows the code to help with caption analysis.

6.3 Modification

In order for someone to go about the continued development of this project, the files that they would need to interact with most would be in the Src directory. For instance, if they wanted to add functionality to the pure data collection routine, they would need to modify CollectData.py. If they wanted to modify the ability to collect geo data information, they would need to modify CollectGeoDataNew.py.

NOTE: For the code to run, the credentials.txt must be filled out appropriately with a valid Instagram [1] username and password.

7.0 User's Manual

7.1 Setup

The following steps are required to successfully set up the project for proper usage:

1. Download the .zip for this project and extract the contents.
2. Ensure you have installed Google Chrome (version 97[1]). If you don't have this version, you will need to follow the instructions in the footnote.
3. Ensure you have installed Python 3.8 or a similar version.
4. Install selenium (use "pip3 install selenium").
5. Install langdetect (use "pip3 install langdetect").
6. Fill in the credentials.txt file in the auth folder with your Instagram login information.

7.2 Usage

Steps to use the project once installed:

1. To collect results:
    1. Go into the src folder within a terminal.
    2. Then run the file CollectData.py using the call "python CollectData.py <hashtag> <# of pages to parse> <end cursor>" [2], where the items in < > are arguments as defined below.
        - <hashtag> The hashtag you are looking for. This will return JSON objects for the most recent posts that contain <hashtag> in them.
        - <# of pages to parse> The number of pages of API requests you would like to parse. There are roughly 50 posts per page for each API request. Increasing this number will increase the amount of data you collect.
        - <end cursor> Typically, you can leave this empty. If you would like to continue a previous search, plug in the end cursor found in the metadata folder in the file corresponding to the Unix time when the previous search was performed.

.

[1] If your version of Google Chrome is 96 or 98, visit https://chromedriver.chromium.org/downloads and replace the executable in the home directory with the one that is compatible with your version of Chrome and ensure that its name exactly matches the current version.

[2] Only the input parameter is required. The default number of pages to parse is 20.

2. To analyze the data:
    1. Ensure the file "data_thinned.csv" is in the top level of the /data/ folder (or replace the one there with a file containing the data you wish to analyze).
    2. Call "python Analyze.py" from your terminal, while in the src directory.
    3. Check the results folder for the results of your analysis (a results folder specific to your query will be there with the Unix time of your analysis, if you selected the option to print the results).

3. To visualize the connections between tags:
    1. Open Visualize.py and locate the main method near the bottom of the Python file.
    2. Manipulate the value of OUTPUT_FILENAME to be the desired filename of your results.
        - i. Note: This string value need not contain any file extensions; it should only be unique relative to the filenames in the results folder.
    3. Manipulate the values in sourceTags to be what you desire.
        - i. Note: sourceTags affect what data is collected (e.g., making the value [‘#appalachiantrail’, ‘#lnt’] would ensure only posts containing one of the two tags in the list are collected for visualization).
    4. Manipulate the values in interestingTags to be what you desire.
        - i. Note: interestingTags affects what connections are drawn for the visualization (e.g., making the value [‘#leavenotrace’, ‘#atclassof2021’] would ensure the visualization only displays connections between tags involving #leavenotrace and/or #atclassof2021).
    5. Run “python Visualize.py” in your terminal from the src directory.
    6. Select “[1]Show connected graph” by entering “1” into the terminal.

4. To visualize the data collected across the years 2015-2021 on a map:
    1. Run “python Visualize.py” in your terminal from the src directory.
    2. Select “[2]Visualize geo data” by entering “2” into the terminal.

8.0 Lessons Learned

8.1 Timeline

- January 25 - Team introductions and role establishments
  - Team selects project based on interests in Instagram [1], hiking, and data analytics.
  - Determine different roles within the team based on varying interests and skills.
  - Meet with sponsors to determine initial goals and expectations.

- February 8 - Data Scraping and presentation one
  - Download Selenium [4], Chrome driver [15], and required dependencies.
  - Create Python [3] scripts to scrape data from Instagram and save data into .txt and .csv files.
  - Have a meeting prior to presentation 1, to discuss our work completed and plans for the rest of the semester.

- March 1 - Finalize data scraping and start sentiment analysis
  - Clean data to get rid of unnecessary posts, such as posts not in English
  - Ensure that the data scraped is satisfactory to the sponsors
  - In addition to having files with all the data together, create files with the data separated by hashtag
  - Start collecting geolocations of different Instagram posts
  - Research different sentiment analysis techniques and make a decision

- March 24 - Finalize sentiment analysis and presentation two
  - Utilize VADER [10] to perform sentiment analysis on the Instagram posts
  - Sentiment scoring is based on key words in the sentence and their mean valence scores
  - Have meeting prior to presentation 2, to ensure each member knows what to talk about and verify that the PowerPoint file is fully polished

- April 7 - Start data visualizations and finish geo collection
  - Determine different viable data visualizations and check with sponsor for approval
  - Finish collecting geolocation data of Instagram posts
  - Implement map of the world with geolocations of Instagram posts overlaid on top
  - Implement visualization of connections between hashtags

- April 14 - Check in with sponsor and final presentation
  - Have meeting with sponsor to determine goals for the rest of the semester and update them on work completed
  - Work on final presentation early, so we are not rushing to finish everything at the end of the semester

8.2 Problems and Solutions

- Python [3] libraries did not allow us to scrape Instagram [1] data

  Since Instagram requires login credentials after less than a dozen requests, we could not use Python libraries to scrape the large number of posts that we were attempting to collect. To solve this issue, we downloaded and used Selenium [4], to create an instance of Google Chrome [5], which allowed us to scrape the data. The login information is passed into the browser as cookies.

- Scraping geolocations of Instagram posts proved time consuming

  After completing the code which retrieves the geolocation of a post, we approximated that it would take around 7 days to scrape all of the geolocations. Our team did not want to use our personal computers for this, as we had other school work to complete. We went to Dr. Fox for advice, and he gave us access to a desktop computer. We have used this computer, alongside Tashi's computer, to scrape all of the data at a much faster rate.

- Maintaining effective time management skills and communication

  At the beginning of the semester, we struggled with time management and communication issues. We would sometimes get behind on work due to these issues. To solve this, we decided to use Discord to keep each other updated more frequently and Ally.io to have a visual way of keeping track of our progress throughout the semester. We also started having each team member say what they did for the week during the weekly meetings with our sponsors. Our communication also improved when any of our team members were struggling with a task. When brought up, the other team members would offer assistance in order to help out the teammate and the progression of the overall project.

8.3 Future Work

8.3.1 Different Method of Scraping the Instagram Data

We relied on using deprecated API calls through Selenium [4] to collect the different Instagram [1] posts. If this API ever becomes defunct, so would our codebase. To avoid this and future proof the code that we have written, future developers could attempt to get a Facebook developers account. We decided not to go down that route as it can potentially cost money and a decent amount of time. However, since Facebook owns Instagram, having a Facebook developer account would make scraping Instagram posts easier.

8.3.2 Different Sentiment Analysis Technique

As mentioned earlier in the paper, we utilized VADER [10] to get the sentiment analysis of each Instagram [1] post. Future developers could try building their own sentiment analysis tool, or could use a different sentiment analysis technique. It could be interesting to compare the results of a different technique to the results we obtained this semester.

8.3.3 Using Current Code Base to Collect Twitter Posts

Although there were more than enough Instagram [1] posts related to the hashtags we were looking for, having more data is usually better. Collecting similar data from Twitter posts and then comparing it to our current results could be useful. The entire code base would have to be refactored to scrape from Twitter, instead of Instagram. However, the logic to do so would be similar and Twitter scraping is extremely popular, so there are several tutorials and help guides available online.

8.3.4 Continuing the K-Means Analysis

Much of the work to be done for the K-Means analysis was left undone, largely due to the sponsor's shifting focus away from this work from our group's weekly meetings. As such, the work left for a future team shall be detailed here. One should note that the file /src/BagOfWords.py is the one used by /src/Analyze.py, but the file /src/Bag2.py was the file worked on to replace the other. It includes a better structure to it to rerun analyses and introduces means to weight words by frequency for their vectorization. This would allow for finer results as the arrays used in the K-Means algorithm would not strictly be ones and zeros. Basic stemming support was also introduced but not fully implemented. The program can compute the Squared

Sum Error (SSE), which is a useful metric in determining the overall error from the number of clusters. As an optimal number of clusters was never discovered, future work could use both SSE and the elbow technique to ascertain the optimal number of clusters. For those unfamiliar, the elbow technique analyzes the relationship between number of clusters and error. It’s expected for error to continue diminishing as the number of clusters increases; however, there are diminishing returns for increasing the number of clusters. As such, the elbow technique is employed to visually locate the “elbow” of the graph (where the slope stops decreasing as fast) and use that point as the number of clusters.

9.0 Acknowledgements

We would like to thank and acknowledge our professor, Dr. Edward Fox for helping us with our semester project, and for helping us solve our issue of geo-data collection taking a long period of time.

We would also like to thank our main sponsor, Morva Saaty, for being friendly, considerate, and professional. She provided us with important information and feedback during our weekly meetings and helped us stay on track throughout the semester.

10.0 References

[1] Instagram. 2022. Instagram home page. Retrieved April 13, 2022 from https://www.instagram.com

[2] Jeffrey Marion. 2022. The 7 principles - leave no trace center for outdoor ethics. (February 2022). Retrieved April 13, 2022 from https://lnt.org/why/7-principles/?gclid=CjwKCAjw6dmSBhBkEiwA_W-EoJXLlmKPnTeEmDly1AL1d1eStUSRhnvXep1YTFBYTsfxDDMwnjRWkxoCkooQAvD_BwE

[3] Python. 2022. Python home page. Retrieved April 13, 2022 from https://www.python.org/

[4] Selenium. 2022. Selenium home page. Retrieved April 13, 2022 from https://www.selenium.dev/

[5] Google. 2022. Google Chrome download page. Retrieved April 13, 2022 from https://www.google.com/chrome/downloads/

[6] Facebook. 2022. User media. Retrieved April 13, 2022 from https://developers.facebook.com/docs/instagram-basic-display-api/reference/user/media/

[7] Zeke Sexauer. 2022. What is unix time? Retrieved April 13, 2022 from https://kb.narrative.io/what-is-unix-time

[8] Python Package Index - Pypi. 2022. Langdetect 1.0.9. Retrieved April 13, 2022 from https://pypi.org/project/langdetect/

[9] Palo Alto Networks. 2022. What is a denial of service attack (DoS) ? Retrieved April 13, 2022 from https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

[10] Aditya Beri. 2020. Sentimental analysis using vader. (May 2020). Retrieved April 13, 2022 from https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

[11] Saif Mohammad. 2015. Emotion survey - SAIF | homepage. Retrieved April 13, 2022 from https://saifmohammad.com/WebDocs/emotion-survey.pdf

[12] Neal Caren. 2019. Word lists and sentiment analysis. (May 2019). Retrieved April 13, 2022 from https://nealcaren.org/lessons/wordlists/

[13] Umar Farooq, Hasan Mansoor, Antoine Nongaillard, Yacine Ouzrout, and Muhammad Abdul Qadir. 2016. Negation Handling in Sentiment Analysis at Sentence Level. Retrieved April 13, 2022 from http://www.jcomputers.us/vol12/jcp1205-11.pdf

[14] Neal Caren. 2019. Word lists and sentiment analysis. (May 2019). Retrieved April 13, 2022 from https://nealcaren.org/lessons/wordlists/

[15] Chromium. 2022. ChromeDriver - WebDriver for Chrome. Retrieved April 13, 2022 from https://chromedriver.chromium.org/downloads