Geo-Localization Coursework
Web Science | Isaac Tabb | 09-03-2023

Task 1: Geolocalization

(1a) The following algorithm was created to organize tweets into grid boxes of 1km x 1km.

initialize bounding box as:
"top_right": [51.686031, 0.28036],
"top_left": [51.686031, -0.563],
"bottom_right": [51.261318, 0.28036],
"bottom_left": [51.261318, -0.563]

define function to compute distance with one parameter, a list of two sets of coordinates
initialize latitude1, longitude1 from passed in param.
initialize latitude2, longitude2 from passed in param.
using the coordinates, compute Haversine's formula for distance on sphere
return Haversine's formula value

initialize height (in km) of London as distance between top-right point and bottom-right point
>> by calling compute distance function and passing points as list
initialize width (in km) of London as distance between top-right point and top-left point
>> by calling compute distance function and passing points as list

calculate number of grid boxes as rounded height and rounded width (in km)
>> ensuring that no grid is less / greater than 1km by 1km

define function which assigns tweets to grid boxes with two parameters, tweet and bounding box
initialize tweet coordinates as value in 'coordinates' of json load
calculate distance between (tweet latitude, top-left point longitude) and the top-left point of the bounding box
>> the math.floor of this distance is the grid box the tweet is in width-wise
calculate distance between (top-left point latitude, tweet longitude) and the top-left point of the bounding box
>> the math.floor of this distance is the grid box the tweet is in height-wise
assign tweet to this grid box (height box, weight box)

iterate over all tweets in list
calculate grid box for each tweet by calling function

(1b) The dataset contains a total of 10526 tweets. Figure 1 is a table of the top-10 grid boxes by number of tweets contained in them. The grid box with the largest number of tweets was (19, 31), or the 19th row and the 31st column. The (19, 31) box contained 34.2% of all tweets in the entire dataset and was located in the center of London between the City of London & Soho. Notably, the remainder of the top ten boxes are all within four rows or columns of the (19, 31) box. Thus, the top ten boxes show that the vast majority of tweets in the dataset are located very close to the center of the city. To paint a clearer picture of the distribution of tweets, **Figure 2** depicts the number of tweets within ***x*** rows or columns of the (19, 31) box. Figure 2 shows that nearly 75.9% of the tweets in the entire dataset are located within at most three rows or columns of the hottest box, (19, 31). In other words, over three quarters of the geo-tagged tweets are located relatively close to the center of London. Additionally, there were no tweets that came from seventeen or more rows/columns away from the hottest box. Not only are the majority of the tweets coming from the center of London but there are little to no tweets that came from the outskirts of the bounding box. The data is concentrated heavily in the center of the city. One might think that this concentration is because the majority of people who live in London live in the center of the city, but the percentage of tweets is an overrepresentation of how many people actually live in the middle of London. According to Trust For London[1], only 16% of London's population lives in Central London (as of 2021). Even by combining Eastern London (the largest sub-population) with Central London, the percentage is still only 48.2%, far less than the 75.9% obtained from the tweets. Tourism could be one explanation for the disproportionate number of tweets coming from the center of London. Tourists tend to spend time where London's attractions are located. If tourists are contributing to a large portion of the tweet data, this might explain why a large number of tweets are located near the center of the city.

Figure 1: Top-10 Grid Boxes

Rank	Grid Box	# of Tweets
1	(19, 31)	3605
2	(19, 30)	981
3	(19, 29)	564
4	(23, 33)	479
5	(19, 32)	401
6	(20, 30)	186
7	(19, 28)	168
8	(21, 34)	150
9	(18, 30)	135
10	(18, 28)	128

Figure 2: Distance from (19, 31) Box

Within ***x*** Grid Boxes of (19, 31)	# of Tweets
x=1	5630 (53.5%)
x=2	7078 (67.2%)
x=3	7984 (75.9%)
x=4	8895 (84.5%)
x=5	9434 (89.6%)
… x=17	0 (0.0%)

(1c) To visualize the distribution of the data, Figure 3a depicts a heat map of the distribution of tweet locations, where darker blues indicate higher numbers of tweets and greens indicate little to no tweets. As noted in (1b), the vast majority of the tweets are located in the center of the heatmap. Figure 3b depicts a zoomed in version of the heatmap, focusing on the areas within ten rows or columns of the hottest box, (19, 31).

[1] https://www.trustforlondon.org.uk/data/geography-population/

Figure 3a: Distribution Heatmap

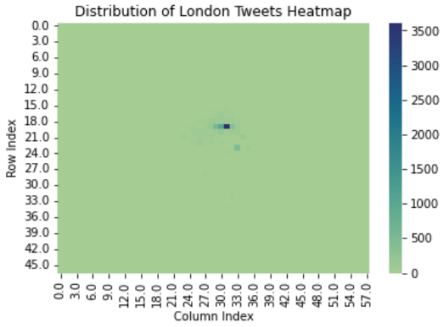
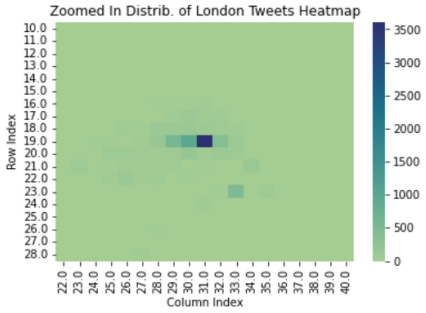
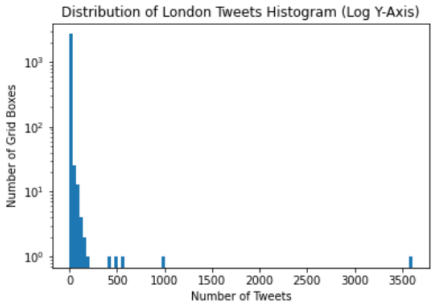


Figure 3b: Distribution Heatmap (Zoomed)



The above heat maps clearly depict the overrepresentation of tweets in the center of the city. The only grid box that is noticeably dark in the entire map is the (19, 31) box, while the other shaded boxes are only a light blue-green. Figure 4 illustrates a histogram which similarly depicts the skew of the distribution. To best show the distribution of grid box sizes, a log-scale y-axis was employed. The skew of the distribution is clear when looking at the sheer number of grid boxes which contain zero tweets compared to the single grid box which has over 3500 more tweets.

Figure 4: Distribution Histogram



(1d) At first glance of the geographic distribution of the tweets, one may think that the majority of tweets that are posted from London are sent from the center of the city. The hottest grid box, (19, 31), located between Soho and the City of London contains 34.2% of the data just by itself (see Figure 1). Additionally, 89.2% of the tweets are located within five grid boxes of the hottest box, (19, 31) (see Figure 2). When taking a closer look into the data though, one will notice an interesting trend with the geo-coordinates provided in the dataset. Although the tweets within the (19, 31) grid box have geo-location enabled, 97.8% of the 3605 tweets in this box are located at the exact same coordinates: (-0.1094, 51.5141) (see Fig. 13 / on p.12). The reason that such a large proportion of these tweets are located at this exact position is because they all come from the same source, Instagram. Twitter provides an attribute called ‘source’ which lists the utility which was used to post the tweet. When looking into the distribution of the sources for each tweet, one will find that 76.2% (8019) of all of the tweets in the dataset were posted via Instagram (see Fig. 12 / on p.12). Another 10.3% (1083) were posted using Career Arc 2.0, a remote career platform. In fact, only 1 of the 10526 tweets in the dataset was posted from source ‘Twitter for iPhone’ and the rest come from outside sources. Thus, using the coordinates provided in this dataset likely does not result in a very accurate organization of tweets into grid boxes since all of the tweets are coming from outside sources. The accuracy of location data from outside sources can vary, thus outside applications cannot be used as trusted sources for

fine-grained geo-location data. The (19, 31) grid box is a prime example of outside sources assigning generalized coordinates to tweets. All of the tweets from Instagram, no matter the exact location they were sent from, were given the same coordinates.

Task 2: Newsworthiness Computation

(2a) The newsworthiness computation utilizes all three of the models: High Quality (HQ), Low Quality (LQ), and Background (BG). Before computing the newsworthiness score, some preprocessing steps were employed. First, the HQ and LQ models were split into training and validation sets. This made it so that data analysis could be conducted on a validation set that was not used to construct the newsworthiness computation. Using the Spacy library, the HQ and LQ tweets were tokenized. Both the tweet text and user description were tokenized and joined together into a token list.

After tokenization, the term frequencies for each JSON object were computed along with the sum of term frequencies for each model. In the baseline scoring method, the quality score of each tweet was utilized to weight the term frequencies. For instance, if an HQ tweet had a quality score of 0.75 and one occurrence of term T, the frequency of term T would be (1 * 0.75), or 0.75. On the other hand, if that tweet had a score of 0.65, the term would only have a frequency of 0.65. Theoretically, this would make it so that terms that occur in the *highest* quality tweets are weighted more in the HQ model. Similarly, for the LQ model, the weighting used was *one minus* the quality score, making it so that the *lowest* quality tweets were weighted more in the LQ model. Since tweets in the BG model also include a quality score, these tweets were weighted by their quality score as well. The average LQ quality score was 0.421 while the average score for HQ tweets was 0.625. Theoretically, an "average" tweet which is neither low nor high quality would have a score of 0.5. So, to ensure a fair comparison of LQ terms to HQ terms, all of the LQ quality scores were scaled so that the average quality score was 0.375. Thus, both the LQ and HQ average quality scores were 0.125 points away from the "average" tweet score of 0.5. When analyzing how the newsworthiness computation performed on the validation set though, there was reason to believe that the quality score was not actually beneficial for the model (see 2b). So, the final scoring method did not include the quality score and instead used raw term frequencies.

Once preprocessing was complete, the relative importance of each term in each model was computed. In the instance of the HQ model, the relative importance was calculated as the probability of term T in HQ divided by the probability of term T in the BG model, and similarly for the LQ model. Since some terms in the HQ and LQ models did not appear in the BG model, add-0.5 smoothing was employed on every term frequency in each model (including BG as well). This prevented any terms from having a probability of 0 in the BG model. The final newsworthiness score was then calculated by iterating through the tokens of a tweet. For each

token, if that token had a relative importance greater than 2.5 in the HQ model, that importance was added to the tweet's HQ sum. The same was done for the LQ model. The final score was then computed as the log base-2 of (1 + HQsum) / (1 + LQsum). Scores greater than 0 were considered HQ and scores less than 0 were considered LQ.

Newsworthiness Computation Pseudocode:

Initialize Background Model TF dictionary as empty
Initialize the total number of terms in the Background Model as 0
Iterate through the Background Model JSON object list
Use collections.Counter to save token counts for each object as dict
Iterate through key and item pairs in dictionary
If the key is in the Background TF dict
Increment the value at that key by the TF of that key in the current tweet
Increment the total term freq. in BG model
Else the key is not in the BG TF dict yet
Assign that key in the dict as the TF of that key in the current tweet + 0.5
Add 0.5 to apply smoothing
Increment the total term freq. in BG model

Define a tokenization method for the High Quality and Low Quality models
(Using Spacy)
Iterate through the tweet contents
Remove stopwords, punctuation, spaces
Convert to lowercase and lemmatize
Iterate through the resulting tokens
If a token includes '•' (which is not recognized by Spacy)
Remove that character from the token
Return the list of tokens

Iterate through tweets in High Quality model
Tokenize the tweet text
Tokenize the tweet description
Combine into list of tokens
Iterate through the tokens
Remove links and non-alphabet terms from tokens

Initialize a stopwords list
Initialize HQ Model TF dictionary as empty
Initialize the total number of terms in the HQ Model as 0
Iterate through the HQ Model JSON object list
  Use collections.Counter to save token counts for each object as dict
  Iterate through key and item pairs in dictionary
    If the key is a stopword (this happens because of '•')
      Pass as to not keep in token list
    Else If the key is in the HQ TF dict
      Increment the value at that key by the TF of that key in the current tweet
      Increment the total term freq. in HQ model
    Else the key is not in the HQ TF dict yet
      Assign that key in the dict as the TF of that key in the current tweet + 0.5
        Add 0.5 to apply smoothing
     Increment the total term freq. in HQ model

Repeat the same preprocessing for the LQ model as for the HQ model

Initialize dict to hold significant importance terms in HQ model
Iterate through the key item pairs in the HQ TF dictionary
  Try calculating the relative importance as:
    Relative Importance = Prob(Term in HQ) / Prob(Term in BG)
  Except if Term does not exist in BG model calculate as:
    Relative Importance = Prob(Term in HQ) / (0.5 / Total terms in BG)
  If the relative importance is greater than 2.5
    Assign relative importance to Term in HQ relative imp. dictionary

Repeat the same relative importance computations for the LQ model

Define score method which takes HQ & LQ relative importances along with tweet
  Initialise sHQ, sLQ as 0
  Iterate through tokens in tweet
    Try increment sHQ for current token

Except pass (this means sHQ was 0)

Try increment sLQ for current token

Except pass (this means sLQ was 0)

Compute newsworthiness score as log base 2 of (1+sHQ) / (1+sLQ)

Return newsworthiness score

(2b)

Figure 5a: Quality Score / No Stopwords / Threshold = 2 (QNST2)

	Accuracy	Mean Score	Std. Dev.
HQ	0.780	1.613	1.906
LQ	0.941	-4.620	4.067

Figure 5b: Quality Score / Keep Stopwords / Threshold = 2 (QST2)

	Accuracy	Mean Score	Std. Dev.
HQ	0.504	0.021	0.350
LQ	0.928	-0.867	0.781

Figure 5c: No Quality Score / No Stopwords / Threshold = 2 (NQNST2)

	Accuracy	Mean Score	Std. Dev.
HQ	0.832	1.856	1.997
LQ	0.925	-4.646	4.113

Figure 5d: No Quality Score / No Stopwords / Varying Thresholds (NQNSTx)

	Signif. Terms	Accuracy	Mean Score	Std. Dev.
HQ (Threshold = 1.5)	6403	0.830	1.807	1.942
HQ (Threshold = 2.5)	2306	0.818	1.836	2.057
HQ (Threshold = 3.5)	2207	0.816	1.899	2.140
HQ (Threshold = 4.5)	1251	0.802	1.938	2.246
LQ (Threshold = 1.5)	1932	0.925	-4.623	4.096
LQ (Threshold = 2.5)	1630	0.928	-4.715	4.154
LQ (Threshold = 3.5)	1516	0.915	-4.696	4.231
LQ (Threshold = 4.5)	1467	0.921	-4.744	4.207

During the process of building the scoring method, multiple different features were tuned and evaluated to see which scoring method performed best. The baseline scoring method, QNST2 (see Figure 5a), was incredibly accurate in predicting that LQ tweets were non-newsworthy (94.1%), but performed less well on the HQ dataset, predicting only 78.0% of the HQ tweets correctly. Additionally, QNST2 gave the LQ tweets an average score of -4.620, which was very promising. For the HQ tweets though, the average score was 1.613, which while that value is significantly above zero, it did not translate to great accuracy in predicting HQ tweets as newsworthy.

The second method that was evaluated similarly used quality score weighting and a threshold of 2, but this time kept stopwords in the HQ and LQ token sets (see Figure 5b). Keeping stopwords proved to have a negative impact on both the HQ and LQ models but especially on the HQ model, reducing the prediction accuracy to 50.4%. The poor performance by QST2 on the HQ model can be attributed to the frequency of stopwords in the HQ model as opposed to in the LQ model (see Figure 6). The top terms in the HQ model were dominated by stopwords (with the exception of

Figure 6: Top Terms (QST2)

High Quality	Low Quality
the: 1032.35	the: 486.99
london: 969.75	a: 443.72
and: 686.70	mohammed: 439.49
in: 631.18	al: 413.76
of: 611.66	nasser: 404.36
a: 584.24	yamani: 386.78
be: 576.08	imam: 371.73
to: 537.80	be: 362.86
at: 398.19	almahdicaliphofallah: 359.83
i: 372.43	and: 349.64

'london'), each with an incredibly high weight. There were no stopwords in the BG set, so these high weights also resulted in high relative importances. Since stopwords had the highest relative importances, the HQ tweets in the QST2 model became *defined* by their stopwords. Since stopwords occur in almost all tweets, HQ tweets became indistinguishable. On the other hand, the LQ model was not defined by stopwords as only four of the top-10 weighted words were stopwords. Thus, there were still heavily weighted terms in the LQ model that could help to distinguish an LQ tweet's lack of newsworthiness.

Figure 7a: Top Terms in HQ and LQ Validation Sets

High Quality	Low Quality
London (370.5)	Mohamed (149.5)
Kingdom (142.5)	Nasser (149.5)
United (129.5)	Al (149.5)
World (90.5)	Yamani (142.5)
Photo (67.5)	Almahdicaliphofallah (135.5)

Since keeping stopwords did not improve performance, the next method that was employed, NQNST2 (see Figure 5c), tried taking away the quality score from the weighting of terms. Without the quality score in the weighting scheme, 83.2% of the HQ tweets were classified as newsworthy, an improvement of approximately 6 percentage points in comparison to the quality score method. For the LQ tweets, the result was only slightly worse as the accuracy dropped by 2 percentage points to 92.5%. The reasoning for these changes is likely a result of the ambiguity of high quality tweets. When examining the top-5 terms from the HQ and LQ validation sets (see Figure 7a), one can see that the top terms for the HQ set do not distinguish high quality London tweets from all tweets about London. The top three terms in the HQ set were 'London', 'Kingdom', and 'United'. On the other hand, the LQ top terms are far more specific. Since HQ tweets are hard to distinguish from London tweets as a whole, the quality scores for HQ tweets might not be incredibly accurate. This may explain why including the quality score hurts the accuracy of predicting HQ tweets.

Since using the quality score was ineffective and keeping stopwords heavily impacted HQ tweets, it was clear that NQNST2 was the best option for the newsworthiness computation. Since the original threshold had been chosen somewhat arbitrarily at the value of 2, the next step was to test varying thresholds to see which performed best (see Figure 5d). Of the five thresholds (including the baseline of 2), threshold's 1.5 and 2 were the most effective. Both thresholds yielded accuracies of ~83.0% on the HQ set and 92.5% on the LQ set. The threshold of 2.5 was also effective with an accuracy of 81.8% on the HQ set and 92.8% on the LQ set. The remaining thresholds slowly became less effective as their size increased.

Before choosing a threshold for the final scoring method, it was important to analyze why it might be that lower thresholds performed better. An interesting pattern arises when looking at the top terms in the training and validation sets of the HQ and LQ models (see Figure 7b). Of the top-10 weighted terms in the HQ training set, nine of those ten terms also appeared in the top-10 of the HQ validation set, just in a different order. Even more interestingly, of the top-10 terms in

Figure 7b: Top Terms (Training vs. Validation)

Rank	HQ Training	HQ Validation		LQ Training	LQ Validation
1	London	London		Mohamed	Mohamed
2	Kingdom	Kingdom		Al	Nasser
3	United	United		Nasser	Al
4	World	World		Yamani	Yamani
5	Official	Photo		Imam	Almahdicaliphofallah
6	Photo	Post		Almahdicaliphofallah	Imam
7	Meet	Britain		London	London
8	Britain	Meet		Kingdom	Kingdom
9	Post	Official		United	United
10	Handle	Like		Post	Post

the LQ training set, all ten of those terms appeared in the top-10 of the LQ validation set, nearly in the same order. Since the training and validation sets were partitions of shared larger sets, the pattern of similarity between the training and validation sets is likely not a coincidence. The training and validation sets are probably very similar. Lowering the threshold for relative importance includes more features from the training set, thus lowering the threshold is similar to overfitting on the training set. Since the validation set looks very similar to the training set, lowering the threshold will lead to better performance on the validation set. Although the performance of lower thresholds might lead one to believe that they are more effective, if a low-threshold scoring method was given an unfamiliar tweet, it would likely perform poorly.

The threshold that was chosen for the final scoring method was 2.5 since it yielded strong accuracy scores without being too small. Thus, the final scoring method does not employ the quality score, does not keep stopwords, and uses a threshold of 2.5. The final scoring method is very effective in predicting that low quality tweets are not newsworthy (92.8%) (see Figure 5d). The final method is not *as* effective on HQ tweets (81.8%) in comparison to LQ tweets, but it is still very accurate.

Task 3: Applying Newsworthiness to Geo-Localization Data

Figure 8a: Top-10 Terms for Newsworthy & Non-Newsworthy (w/ Threshold = 0)

Term (News)	Frequency	Term (Non-News)	Frequency
London	4815	London	4404
Tweet	2190	Photo	3790
Job	2013	Post	3666
Follow	1226	Kingdom	3068
Need	1142	United	2679
Help	1121	Vape	732
Account	1119	Base	323
Target	1072	Reviewer	308
Geo	1066	Unıted (note: the ı)	275
England	853	Artist	264

Figure 8b: General Newsworthiness Statistics

Mean Score	0.702
Std. Dev. of Scores	1.748
% Newsworthy (>0)	54.7%

(3a) On the data from Task 1, the original threshold applied to define newsworthiness was a threshold of 0. Thus, tweets that had a score greater than 0 were considered newsworthy while those with scores below 0 were considered non-newsworthy. This threshold allowed 54.7% of tweets in the dataset to be considered newsworthy (see Figure 8b). Looking at the top-10 newsworthy and non-newsworthy terms (see Figure 8a), the first thing that stands out is that the vast majority of tweets in both groups are about London. When it comes to differences between the two groups, the newsworthy group appears to have a focus on service and goals as we see terms like “job”, “need”, “help”, and “target” at the top. On the other

hand, the tweets in the non-newsworthy group are more about media and promotion with terms like “photo”, “post”, “base”, “reviewer”, and “artist”. Notably, the term “vape” also appears a lot which makes sense for a non-newsworthy tweet.

The ratio of high quality term weight to low quality term weight in the newsworthy tweets was also analyzed to examine how *definitively* newsworthy the tweets were. Using the threshold of 0, the ratio of high quality weight to low quality weight in newsworthy tweets was only 2.46:1 (see Figure 8c), meaning that newsworthy tweets only had 2.46 times more high quality term weight than low quality term weight. Although that ratio leans towards newsworthiness, it does not indicate that the average newsworthy tweet is *conclusively* newsworthy. A ratio of 3.5 or 4 to 1 would be a stronger indicator that the average newsworthy tweet is far more newsworthy than it is non-newsworthy. To attempt to raise this ratio, the threshold for newsworthiness was moved from 0 to 0.702, with 0.702 being the average newsworthiness score across the entire dataset (see Figure 8b). The intuition behind this is that newsworthy tweets should be above the average newsworthy score for all tweets. When the threshold was raised to 0.702, the ratio of high quality weight to low quality weight increased to 4.21:1 (see Figure 8c). A ratio above 4:1 indicates that the average newsworthy tweet has considerably more high quality weight than low quality weight. Thus, the final choice for threshold was 0.702.

Figure 8c: Ratio of HQ : LQ Weight by Threshold

Threshold	Ratio (HQ:LQ)
>0	2.46HQ : 1LQ
>.702 (Mean)	4.21HQ : 1LQ

(3b) After removing tweets below the threshold of 0.702, only 4094 (38.8%) of the original 10526 tweets remained and were considered newsworthy (see Figure 9a). This means that 6432 of the tweets, or 61.2%, were deemed non-newsworthy and removed. The average score of a newsworthy tweet was 2.54, a high enough score to once again indicate that the average newsworthy tweet was *definitively* newsworthy. The distribution of scores for the newsworthy tweets (see Figure 9b) shows that the vast majority had scores greater than 1.0. Nearly 60% of the newsworthy tweets had scores greater than 2.0. Even at the threshold of 3.0, still nearly 40% of all newsworthy tweets were above the threshold. There were not a significant number of tweets with incredibly high scores though, with only 13.2% of tweets having scores above 4.0. That percentage dwindles even more as the thresholds get higher.

Figure 9a: General Newsworthy Tweet Statistics (> 0.702)

Tweet Count	4094 (38.8%)
Mean Score	2.54

Figure 9b: Distribution of Newsworthiness Scores

Threshold	Count (Percentage)
w/ Score > 1.0	3523 (87.2%)
w/ Score > 2.0	2413 (59.7%)
w/ Score > 3.0	1548 (38.3%)
w/ Score > 4.0	535 (13.2%)
w/ Score > 5.0	172 (4.3%)
w/ Score > 6.0	24 (0.6%)
w/ Score > 7.0	2 (<0.1%)

Figure 9c: Newsworthiness & The (19, 31) Grid Box

	(19, 31)	(19, 30)	All Non-(19, 31) Grid Boxes
% Newsworthy	7.5%	72.3%	55.2%
% Non-Newsworthy	92.5%	27.7%	44.8%

When looking at the distribution of **location** for newsworthy and non-newsworthy tweets, an interesting pattern arises. For the newsworthy tweets, the top grid box was (19, 30), where 709 (72.3%) of the newsworthy tweets were located (see Figure 9c). For the non-newsworthy tweets though, the top grid box was (19, 31), which referring back to Task 1 was by far the top grid box overall. Out of the 3605 tweets located in the (19, 31) grid box, 3335 (92.5%) of them were considered non-newsworthy. Notably, 3335 is 51.9% of all non-newsworthy tweets in the entire dataset. When looking even closer, out of the 6921 tweets not located in the (19, 31) grid box, 55.2% of them were considered newsworthy. In other words, the majority of non-newsworthy tweets are coming from the (19, 31) grid box, which is interestingly located in the center of London. On the other hand, newsworthy tweets appear to be a lot more spread out across the greater London area.

(3c)

Figure 10a: Newsworthy Distribution Heatmap

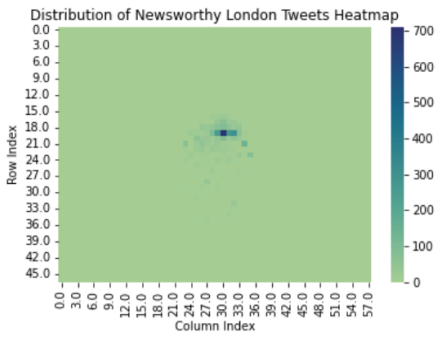


Figure 10b: Non-Newsworthy Distrib. Heatmap

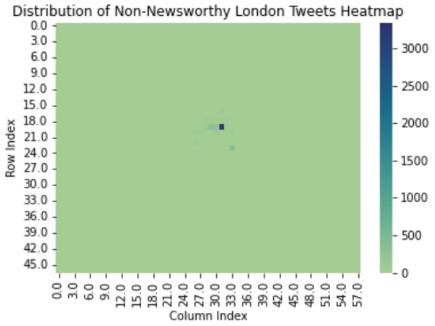


Figure 10c: Newsworthy Distribution Histogram

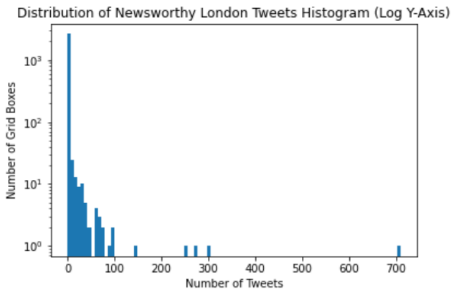
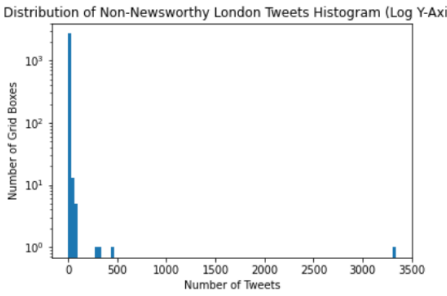


Figure 10d: Non-Newsworthy Distrib. Histogram



Compared to the results in Task 1, one can see that the distribution of non-newsworthy tweets (see Figure 10b) is very similar to the distribution for the entire dataset (see Figure 3a). In Task 1, it was noted that the majority of tweets are located in just a couple of grid boxes in the very center of London, most notably the grid box (19, 31). The vast majority of the non-newsworthy

tweets remained in the center. As noted in Task 1, one explanation for the concentration of tweets in the center of London could be tourism. London tourists tend to spend time in the center of the city since that is where the majority of attractions are located. The tourism explanation holds when looking at the location distribution and top-5 terms of non-newsworthy tweets. In addition to the concentration of tweets depicted in Figure 10b, the top terms for non-newsworthy tweets were clearly related to posting photos in London, an action that is common among tourists (see Figure 11). Looking at the heat map for newsworthy tweets, the distribution is far more spread out. There are significantly more boxes that are shaded slightly blue, indicating that numerous areas of London have newsworthy tweets. This makes sense since news in London should not only be located in the center of the city. It makes sense that there is *more* news in the center of London since central London has a higher population, but it is important that news is also coming from the less populated areas.

Figure 11: Top-5 Non-News Terms (< 0.702)

Rank	Term	Rank	Term
1	london	4	kingdom
2	photo	5	united
3	post		

In terms of the distribution of grid box sizes (see Figure 10c/d), the same pattern arises as seen in the heat maps. For the non-newsworthy tweets, the distribution bar chart looks very similar to the chart for the entire dataset, with a wide majority of boxes having zero tweets. The newsworthy bar chart on the other hand shows that newsworthy tweets are more spread out. There are still a large number of grid boxes with zero newsworthy tweets but there are also far more grid boxes with tweet counts in between 0 and 100. Many of the less populated areas are going to have less news, so it follows that those grid boxes should have a smaller number of newsworthy tweets.

(4)

Figure 12: Top Sources in Dataset

Source	Tweet Count
Instagram	8019
Career Arc 2.0	1083
FourSquare	400

Figure 13: Common Coordinates for (19, 31) & (19, 30)

Grid Box	Coordinates	Tweet Count
(19, 31)	(-0.1094, 51.5141)	3527 (97.8%)
(19, 30)	(-0.1277583, 51.5073509)	546 (55.7%)

When looking deeper into the nature of the tweet sources in this dataset, there is reason to believe that there are issues with performing geolocalization on the data. Although Twitter's geo-enabled coordinates feature is meant to give accurate tweet coordinates, the issue with this data is that the majority of the coordinates are not actually coming from Twitter. Out of the 10526 tweets in the dataset, 8019 of the tweets are coming from Instagram (see Figure 12). When a user posts a tweet using a third-party platform, the coordinates for that tweet are assigned by the third-party platform itself. There is evidence to suggest that Instagram's coordinate system is not as accurate as Twitter's. Out of the 3605 tweets in the (19, 31) grid box, 97.8% of those tweets had the exact same coordinates of (-0.1094, 51.5141) (see Figure 13). When looking at the sources of these tweets, all of them came from Instagram. Instagram is likely assigning the same general location for all people who are sending tweets from that area of

London. When looking at the second most popular grid box, (19, 30), a similar pattern arises. Out of the 981 tweets in (19, 30), 546 of them are from the coordinates (-0.1277583, 51.5073509). The majority of these tweets are from the source, CareerArc2.0, while a few others come from Squarespace and Instagram. One can infer that these platforms are likely choosing a specific coordinate point to represent all tweets that are sent from the same general area of London.

Since the coordinates that are assigned by outside sources are not incredibly accurate, they cause issues for geolocalization. For instance, if a local business located in the (19, 30) grid box wanted to use these tweets to identify potential users to market to, the inaccuracy of the coordinates could cause the business to waste resources on users who are not actually located nearby. If a major event was occurring in the center of London, news sources might want to gather Twitter data on first-hand accounts of the event. If a dataset like this was used though, the coordinates provided would not be able to accurately pinpoint people who were at the scene of the event when it happened. In short, using geo-localization makes the assumption that the locations provided are fine-grained and highly accurate. If the accuracy of the coordinates provided cannot be trusted, then the data is of no use for analysis.