Technical Document

Anger Measurements in Twitter Data

Project conducted by:

San Diego State University

5500 Campanile Drive

San Diego, CA 92182-4493

Report prepared by:

Ilana Goldflam

Department of Linguistics, San Diego State University

January 2019

Abstract

This technical document discusses applying the anger words to Twitter data collected about any topic or event. This provides a relative measure of anger overall. It can also be used to look at anger over time or to look at the content of angry tweets in a sample, including which words are strongly associated with angry tweets (PMI).

Applying Anger Word Counts to Existing Twitter Data

Introduction

The anger words are keywords and phrases to identify angry tweets with high precision. These anger words are intended as a relative, not absolute, measure of the anger expressed in a sample of tweets.

The anger words are: liar(s), lying, lies, hypocrite(s), hypocrisy, hypocritical, asshole(s), bullshit, fuck AND off, “fuck you”, disgrace(s), disgraced, disgraceful, “piece of shit”, “the fuck up”, piss(ed) AND off, STFU, disgusting, disgusted, disgusts, “go fuck yourself”, scum, infuriate(s), infuriating, infuriated.

This technical document describes four applications of the anger words to Twitter data collected for any keywords.

General Instructions

The following steps require Python and the following packages: numpy, pandas, re, cytoolz, os, collections, sys. Code is available as Jupyter notebooks on the HDMA-Linguistic GitHub repository (https://github.com/HDMA-SDSU/HDMA-Linguistic/tree/master/Anger_Words).

The functions in each notebook take as input an XLSX file or a folder of XLSX files. Each input file must at least have one column containing tweet text with the header TEXT, with each tweet in a separate row. Inputs are case sensitive, so TEXT must be capitalized. (Other columns can also be included but will not affect the results unless otherwise specified.) The notebooks do not include any data cleaning, so this should be added to the code or done beforehand. In particular, filtering tweets to English only is recommended.

If the data is in CSV files, use Excel to convert them to XLSX files. Alternatively, you can change the relevant portions of the code in the notebook. First, change any instances of "df = pd.read_excel(filepath)" to "df = pd.read_csv(filepath)." Second, change any instances of "if filename.endswith(".xlsx"):" to " if filename.endswith(".csv"):"

Note that you do not need to carry out all steps in this section, only those relevant to your analysis. Each notebook functions independently of the others.

Step 1: Counting Anger Word Percentages

The notebook General-Anger-Check.ipynb (https://github.com/HDMA-SDSU/HDMA-Linguistic/blob/master/Anger_Words/General-Anger-Check.ipynb) can be used to count occurrences of the anger words listed above as a percentage of total tweets. Use angertopic(filepath) to compute anger percentages for one file or angertopics(directory) to analyze all files in a directory. For example: angertopic('/Users/ilana/Election.xlsx') or angertopics('/Users/ilana/Election_Data/')

Results will be exported as one CSV for each input file with the same title plus "anger" at the end, e.g. "Election anger.csv" The file contains a list of keywords, raw counts, and percentage (number of occurrences divided by total tweets), as well as the total occurrences of all anger words and the total tweet count, as shown. Note that the percent column is **not** in decimal form; 0.15 means 0.15%, not 15%. Small percentages are typical for these results.

	Counts	Percent
asshole	8	0.14617212
assholes	3	0.05481454
bullshit	28	0.51160241
disgrace	10	0.18271515
disgraced	2	0.03654303
disgraceful	9	0.16444363
disgraces	1	0.01827151
disgusted	4	0.07308606
disgusting	23	0.42024484
disgusts	1	0.01827151
fuck off	5	0.09135757
fuck you	5	0.09135757
go fuck your:	1	0.01827151
hypocrisy	2	0.03654303
hypocrite	3	0.05481454
hypocrites	0	0
hypocritical	2	0.03654303
infuriate	0	0
infuriated	0	0
infuriates	0	0
infuriating	0	0
liar	34	0.6212315
liars	2	0.03654303
lies	67	1.22419149
lying	28	0.51160241
piece of shit	1	0.01827151
piss(ed) off	3	0.05481454
scum	1	0.01827151
stfu	15	0.27407272
the fuck up	3	0.05481454
~Total	261	4.76886534
~Tweet Cour	5473	100

These anger words provide a relative, not absolute, measure of the anger expressed in a sample of tweets. Therefore, they can be used to compare the level of anger between samples (for different keywords, in different locations, or at different times) or to compare against a baseline level of anger for a given population. For example, all tweets in San Diego containing the word "Republicans" on a given day could be compared to tweets containing "Democrats" and/or to a baseline created with tweets collected in San Diego for neutral keywords such as "a," "an" and "the."

A high percentage of individual keywords relative to a baseline can also provide information about the nature of the anger being expressed, e.g. *liar* or *hypocrite* showing value judgments against an individual, or *fuck you* showing interpersonal anger.

Step 2: Calculate Angry Tweets Over Time

The notebook Angry-Tweets-Vs-Time.ipynb (https://github.com/HDMA-SDSU/HDMA-Linguistic/blob/master/Anger_Words/Angry-Tweets-vs-Time.ipynb) contains the function anger_time(filepath), which can be used to calculate the number of angry tweets and total tweets per day in a sample of tweets containing multiple days. This shows changes in anger levels over time in a discussion of a given topic. <u>In addition to the TEXT column in input files, this code requires a column titled CREATED_AT_LOCAL containing the date and time of each tweet.</u> Results will be exported as a csv with "angry tweets" added to the end of the filename.

The first column shows the days included in the sample. TEXT_total and TEXT_angry show the total number of tweets and the number of angry tweets on that day. The ratio is the number of angry tweets divided by the total number of tweets on that day, showing what proportion of tweets are angry. Text % and Angry % show the percentage of the total tweets and the total angry tweets in the sample that were made on that day (i.e. total tweets on that day/total tweets overall) as a decimal.

Use caution when interpreting the ratio, as days with a low number of total tweets may have a high ratio with only a few angry tweets. Consider both the TEXT_angry column (absolute number of angry tweets) and the ratio when identifying days with more anger. Also note that, again, the anger words provide a relative measure of anger, not the total amount of anger in the sample.

Day	TEXT_total	TEXT_angry	Ratio	Text %	Angry %
8/12/17	3257	92	0.02824685	0.22082853	0.39655172
8/13/17	2185	28	0.01281465	0.14814564	0.12068966
8/14/17	1951	17	0.00871348	0.13228015	0.07327586
8/15/17	1636	17	0.0103912	0.11092277	0.07327586
8/16/17	1491	20	0.01341382	0.1010916	0.0862069
8/17/17	1022	19	0.018591	0.06929283	0.08189655
8/18/17	751	8	0.01065246	0.05091871	0.03448276
8/19/17	405	2	0.00493827	0.02745949	0.00862069
8/20/17	231	6	0.02597403	0.01566208	0.02586207
8/21/17	223	0	0	0.01511967	0
8/22/17	281	3	0.01067616	0.01905214	0.01293103
8/23/17	255	7	0.02745098	0.01728931	0.03017241
8/24/17	135	1	0.00740741	0.00915316	0.00431034
8/25/17	160	2	0.0125	0.01084819	0.00862069
8/26/17	98	0	0	0.00664452	0
8/27/17	203	4	0.01970443	0.01376364	0.01724138
8/28/17	131	4	0.03053435	0.00888196	0.01724138
8/29/17	124	1	0.00806452	0.00840735	0.00431034
8/30/17	58	1	0.01724138	0.00393247	0.00431034
8/31/17	50	0	0	0.00339006	0
9/1/17	55	0	0	0.00372907	0
9/2/17	26	0	0	0.00176283	0
9/3/17	21	0	0	0.00142383	0

Step 3: Mark Angry Tweets or Filter Data to Angry Tweets

The Anger-Filter.ipynb notebook (https://github.com/HDMA-SDSU/HDMA-Linguistic/blob/master/Anger_Words/Anger-Filter.ipynb) contains three functions:

1. angerfilter(filepath) saves only the tweets in a dataset that contain at least one anger keyword. This can be used to prepare for further analysis of angry tweets only. Results will be exported as a csv with "angry tweets" added to the end of the filename.

2. angertag(filepath) marks angry tweets with "True" in a new column titled "Anger." A tweet is considered angry if it contains one or more of the anger words. All input data (both angry and non-angry tweets) plus this added column will be exported as a csv with "with tagged angry tweets" added to the end of the filename.

3. angerwordtag(filepath) adds a column called "Anger" containing a list of all anger words in each tweet. Tweets with no anger words will have an empty list in this column. All input data plus this added column will be exported as a csv with "with anger words " added to the end of the filename.

The processfolder(directory, function) function can be used to run any of these functions on a directory of data. For example: processfolder ('/Users/ilana/Tweets/', angertag).

Note that because these are high precision but low recall keywords, this is expected to be only a representative subset of all angry tweets, not every tweet containing anger.

Step 4: Find Words Associated With Angry Tweets Through PMI

Pointwise mutual information (PMI) is a measure of word association. In this case, it shows which words are used more often in tweets containing anger words than in all tweets in a sample. When using word counts, the formula of PMI for some word, w, is: $\log_2 ((w_{anger} * N_{total}) / (w_{total} * N_{anger}))$, where w_{anger} is the uses of the word in angry tweets, N_{total} is the total number of words in the entire sample, w_{total} is the uses of the word overall, and N_{anger} is the total word count for all angry tweets in the sample.

A high PMI score means that words occur more often in angry tweets than in the overall sample, while a low PMI means that words occur less often in angry tweets than overall. PMI scores can give an indication of the causes of anger in a sample.

Download the notebook Anger-PMI.ipynb from Github (https://github.com/HDMA-SDSU/HDMA-Linguistic/blob/master/Anger_Words/Anger-PMI.ipynb). This notebook also requires the Natural Language Toolkit's (NLTK) stopwords.

Use angerPMI(filepath) to calculate PMI for all words, removing stopwords, anger words, and any words that occur fewer than 10 times in angry tweets. This 10 word limit may need to be lowered for small samples or those with few angry tweets, but should not be lower than 5. The line to change to lower the limit is indicated in the notebook.

```
#The limit below may need to be changed to a smaller number for small datasets or those with few angry tweets
freq2 = freq[freq['Angry']>10]
```

Search keywords for the specific data sample will not be removed, so they may need to be removed manually if they appear in the results. Results will be exported as a csv with "anger PMI" added to the end of the filename, showing a list of words sorted from highest to lowest PMI.

Note that PMI is less meaningful for smaller samples (in terms of word count), where differences in word use may be due to random chance. Use caution when interpreting results, particularly in data with a low number of angry tweets.

1. Conclusion

The anger words identify angry tweets with high precision and can be used to investigate anger in Twitter data related to any topic, including looking at anger over time, looking at words associated with angry tweets, or filtering to angry tweets for additional analysis.