The Hashtag-Thread Mashup: How Educators Talk to Each Other in Twitter #Edchat

K. Bret Staudt Willet

11/08/2019

Contents

Get set up

This section loads the data and packages and starts to process the data, but doesn't calculate any statistics or create any results.

Load packages

Load the data

Having completed the steps in the setup.Rmd file, you now have the dataset stored in your local repository and can load it as usual. This project uses Twitter #Edchat data that have been run through the `rtweet` R package, which queries the Twitter API to return the most complete set of tweet metadata available, while also removing deleted and protected tweets. See https://rtweet.info/ for details on `rtweet`.

Data analysis

RQ1. What volume of #Edchat tweets must participants navigate?

Time frame

```
## [1] "Tweets were collected from 2017-10-01 to 2018-06-05 (8.12 months)."
```

Volume of tweeters and tweets

```
## [1] "175474 distinct tweeters created 1111643 unique tweets."
```

Tweets per month per user

```
##   mean  sd median  min     max
## 1 0.78 9.5   0.12 0.12 2636.42
```

Minimal participation. Look at one-time tweeters.

```
## [1] "Tweeters who contributed only one tweet to #Edchat: 96466 (54.97%)"
```

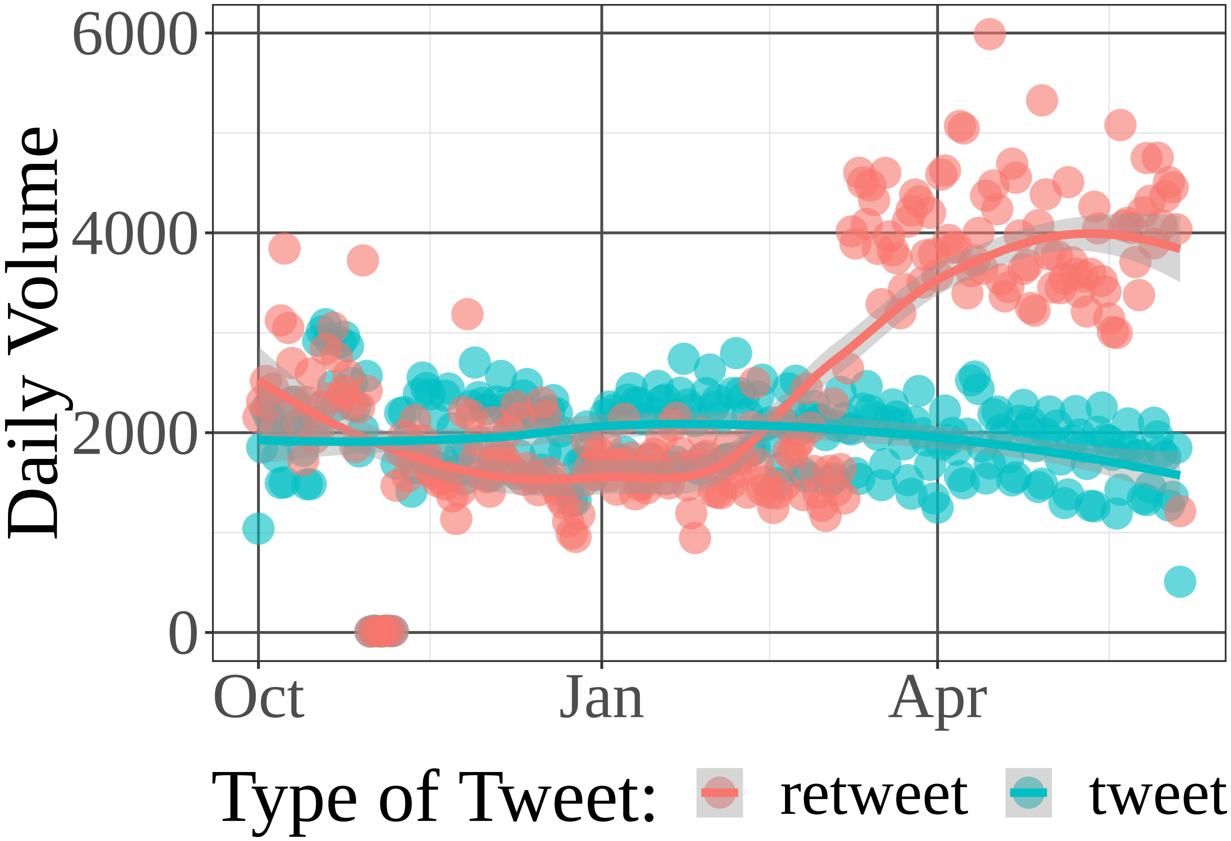
```
## [1] "Tweeters who tweeted at least monthly to #Edchat: 19363 (11.03%)."
```

```
## [1] "Tweeters who contributed by retweeting #Edchat tweets: 162034 (92.34%)"
```

```
## [1] "Tweeters who contributed only one retweet to #Edchat: 93201 (53.11%)"
```

Visualization of daily volume of tweets and retweets

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



RQ2. How do #Edchat tweets demonstrate a mishmash of content, if at all?

First, calculate descriptive statistics for the number of hashtags per tweet:

```
##   mean   sd median min max
## 1 4.13 3.22      3   0  31
```

Now consider how many tweets have #Edchat only and no other hashtags:

```
## [1] "148099 tweets contain the #Edchat hashtag alone (13.32%)."
```

Finally, look at the top-20 hashtags that occur alongside #Edchat:

Hashtag	n
	29875100
#edchat	1116201
#edtech	282393

Hashtag	n
#education	174179
#ukedchat	81285
#elearning	69357
#joyfulleaders	59080
#k12	50467
#leadupchat	47496
#teachers	44540
#mathchat	44412
#engchat	43529
#kidsdeserveit	42174
#teaching	41882
#tlap	38148
#satchat	35555
#aussieed	34367
#learning	29206
#cpchat	26328
#edtechchat	24053
#leadlap	23623
#stem	20589
#highered	19552
#suptchat	18923
#sunchat	18035
#edadmin	17828
#futuredriven	17341
#pbl	16821
#pblchat	16411
#elemchat	15056

RQ3. What social interactions occur within the #Edchat hashtag-thread mashup?

How are participants connected to each other in #Edchat?

```
## [1] "2848 (1.62% of all #Edchat tweeters) replied to someone, in 10381 reply tweets."
```

```
## [1] "1136 (0.65% of all #Edchat tweeters) replied more than once."
```

Replies per replier:

```
##   mean    sd median min max
## 1 3.65 16.29      1   1 637
```

hashtag-thread mashup

Now, reconstruct threads of replies extending beyond #Edchat, starting by looking up tweets that have been replied to but are not in #Edchat.

```
##   mean    sd median min max
## 1 3.59 15.16      1   1 641
```

Volume of tweets in the hashtag-thread mashup

```
## [1] "Overall, the #Edchat hashtag-thread mashup was made up of 13176 replies from 3672 tweeters."
```

```
## # A tibble: 2 x 4
##   has_edchat n_tweets p_tweets n_tweeters
##   <lgl>          <int>    <dbl>      <int>
## 1 FALSE           2743     20.8       1308
## 2 TRUE           10433     79.2       2850
```

```
## [1] "2364 (64.38%) contributors always included the #Edchat hashtag in their replies."
```

```
## [1] "486 (13.24%) contributors sometimes included the hashtag, sometimes not."
```

```
## [1] "822 (22.39%) contributors never included the #Edchat hashtag in their replies."
```

Time Frame

```
## [1] "The #Edchat hashtag-thread mashup ranged from 2014-10-13 to 2018-06-05 (43.75 months)."
```

Create Table 1.

1. Compare the proportion of tweets during weekly *synchronous* chats.
2. Compare the proportion of tweets that are *self-replies.*
3. Look at the difference in average number of *words* per tweet.
4. Look at the difference in average number of *characters* per tweet.
5. Look at the difference in text-polarity *sentiment score* per tweet.
6. Look at the difference in average number of *hashtags* per tweet.
7. Look at the difference in average number of *hyperlinks* per tweet.
8. Look at the difference in average number of *likes* per tweet.
9. Look at the difference in average number of *retweets* per tweet.
10. Look at the difference in average number of *replies* per tweet.

```
## Warning: Each time `sentiment` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory.  It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.

## Warning: Each time `sentiment` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory.  It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.
```

With #Edchat	Sync %	Self-reply %	Words	Characters	Sentiment	Hashtags	Links	Likes	RTs	Replies
TRUE	36.04	15.98	26.53	185.85	0.31	2.51	0.24	2.01	0.48	0.27
FALSE	11.59	29.20	27.90	201.87	0.25	0.57	0.22	26.25	3.66	1.03

Next, create the network graph of #Edchat replies using the `igraph` package.

Then calculate network statistics.

Note that the *diameter* is the length of the longest geodesic (i.e., the maximum distance between two vertices). *Transitivity* is the balance of connections, also called the "clustering coefficient." Transitivity is the probability that the adjacent vertices of a vertex are connected. When the clustering coefficient is large it implies that a graph is highly clustered around a few nodes. When it is low, it implies that the links in the graph are relatively evenly spread among all the nodes (Hogan, 2017). *Reciprocity* is the proportion of mutual connections (in a directed network). That is, reciprocity is the probability that the opposite counterpart of a directed edge is also included in the graph.

```
## [1] "The network of the #Edchat hashtag-thread mashup has 6171 nodes and 13176 edges."
```

```
## [1] "The network has a diameter of 18, a transitivity score of 4.72%,"
```

```
## [1] " and a reciprocity score of 29.19%."
```

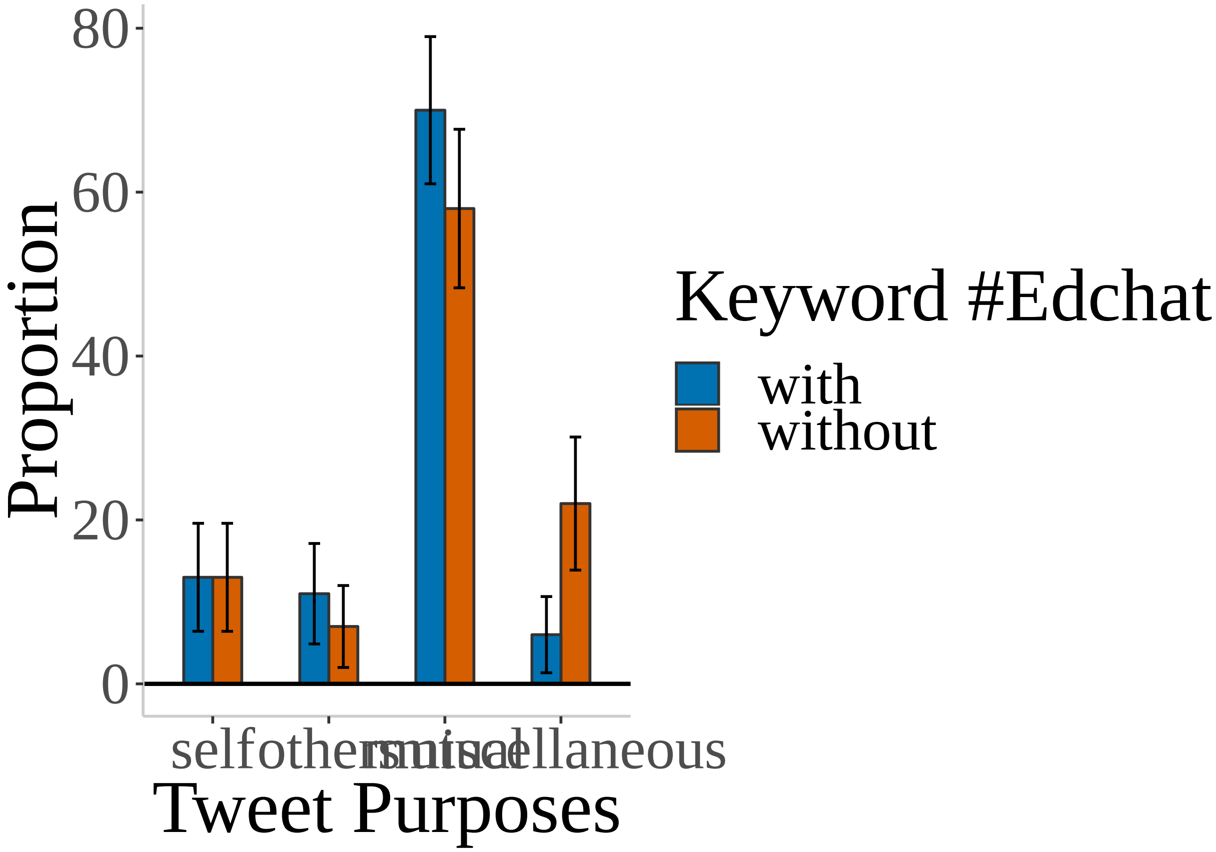
```
## [1] "In addition, the mean degree is 3.47 (SD = 20.15) with a median degree of 1."
## [1] " Finally the minimum degree is 0 and the maximum degree is 1053."
```

Take samples (n = 100) of replies and repliers with keyword #edchat and without.

Visualize these differences in tweet purpose and tweet discourse in replies with and without #Edchat

Purpose

First, calculate margins of error (moe); then display the plot of tweet purposes.



```
## [1] "The two-way table test of association between the presence of the keyword #edchat"
```

```
## [1] "and tweet purpose had a chi-square value of 11.16,"
```

```
## [1] "(df = 3), p = 0.0109 and an effect size d = 0.48."
```

Print the contingency table and the results of the chi-square test.

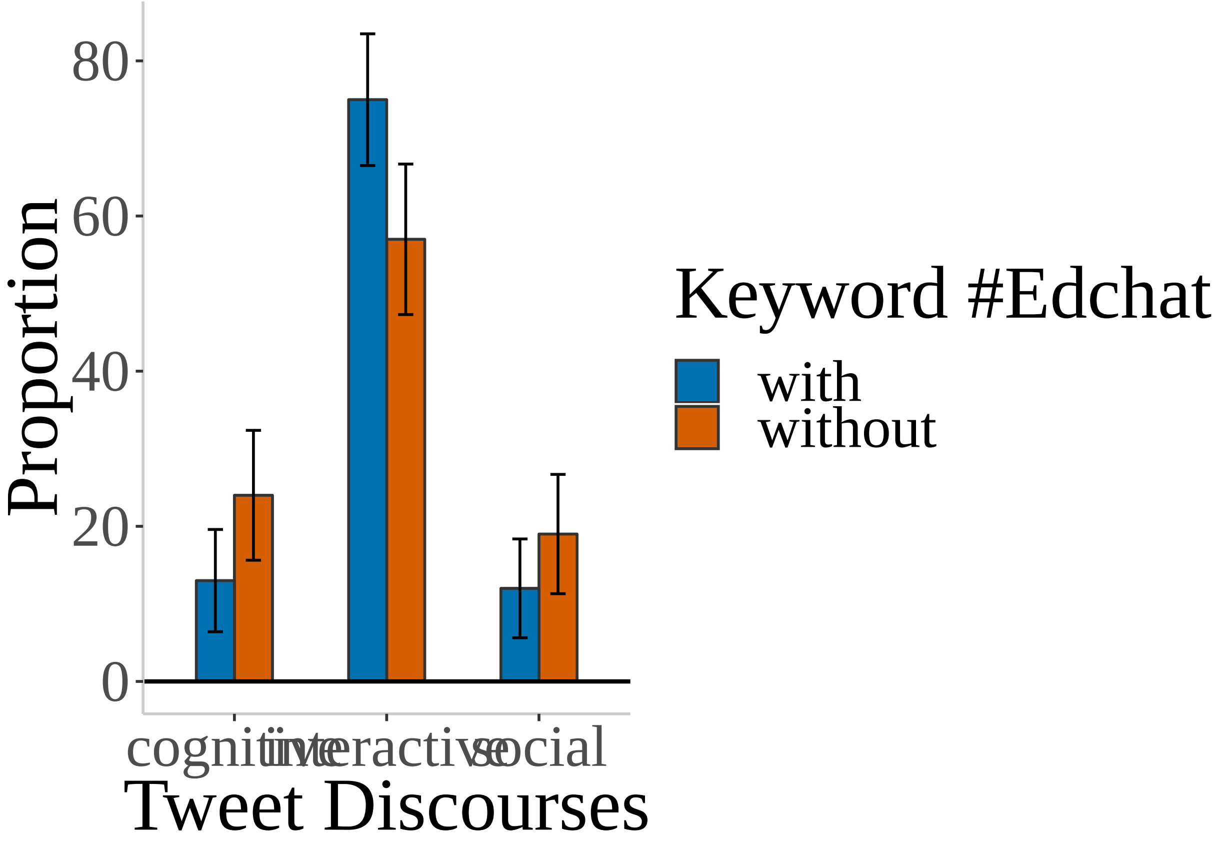
```
##               with without
## self            13      13
## others          11       7
## mutual          70      58
## miscellaneous    6      22
```

```
##
##  Pearson's Chi-squared test
##
## data:  matrix_purpose_sums
## X-squared = 11.157, df = 3, p-value = 0.01091
```

Discourse

First, calculate margins of error (moe); then display the plot of tweet discourses.

Now, display the plot.



```
## [1] "The two-way table test of association between the presence of the keyword #edchat"
```

```
## [1] "and tweet discourse had a chi-square value of 7.31,"
```

```
## [1] "(df = 2), p = 0.0259 and an effect size d = 0.39."
```

Print the contingency table and the results of the chi-square test.

```
##             with without
## cognitive     13      24
## interactive   75      57
## social        12      19
```

```
##
##  Pearson's Chi-squared test
##
## data:  matrix_discourse_sums
## X-squared = 7.3055, df = 2, p-value = 0.02592
```

Version/dependencies

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggraph_1.0.2     igraph_1.2.4.1   sentimentr_2.7.1 rtweet_0.6.8
##  [5] lubridate_1.7.4  forcats_0.4.0    stringr_1.4.0    dplyr_0.8.0.1
##  [9] purrr_0.3.2      readr_1.3.1      tidyr_0.8.3      tibble_2.1.1
## [13] ggplot2_3.1.1    tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] ggrepel_0.8.0     Rcpp_1.0.1        textshape_1.6.0
##  [4] lattice_0.20-38   clisymbols_1.2.0  utf8_1.1.4
##  [7] assertthat_0.2.1  rprojroot_1.3-2   digest_0.6.18
## [10] ggforce_0.2.2     R6_2.4.0          cellranger_1.1.0
## [13] plyr_1.8.4        compute.es_0.2-4  backports_1.1.4
## [16] evaluate_0.13     highr_0.8         httr_1.4.0
## [19] pillar_1.3.1      rlang_0.4.0       lazyeval_0.2.2
## [22] readxl_1.3.1      rstudioapi_0.10   data.table_1.12.2
## [25] textclean_0.9.3   rmarkdown_1.12    labeling_0.3
## [28] polyclip_1.10-0   munsell_0.5.0     broom_0.5.2
## [31] compiler_3.5.2    modelr_0.1.4      xfun_0.6
## [34] pkgconfig_2.0.2   qdapRegex_0.7.2   htmltools_0.4.0
## [37] tidyselect_0.2.5  gridExtra_2.3     fansi_0.4.0
## [40] viridisLite_0.3.0 crayon_1.3.4      withr_2.1.2
## [43] MASS_7.3-51.4     grid_3.5.2        nlme_3.1-139
## [46] jsonlite_1.6      gtable_0.3.0      magrittr_1.5
## [49] scales_1.0.0      cli_1.1.0         stringi_1.4.3
## [52] farver_1.1.0      viridis_0.5.1     fs_1.2.7
## [55] syuzhet_1.0.4     xml2_1.2.0        generics_0.0.2
## [58] tools_3.5.2       glue_1.3.1        tweenr_1.0.1
## [61] hms_0.4.2         yaml_2.2.0        colorspace_1.4-1
## [64] lexicon_1.2.1     rvest_0.3.3       knitr_1.22
## [67] haven_2.1.0       usethis_1.5.0
```