# Introduction

This post provides a brief description of methods for quantifying political bias of online news media based on the media-sharing habits of US lawmakers on Twitter. Here, the focus is on a more streamlined (and multi-threaded) approach to **resolving shortened URLs** via the quicknews package. We also present unsupervised methods for visualizing media bias in two-dimensional space via tSNE, and compare results to the manually curated fact and bias checking online resource, Media Bias/Fact Check (MBFC), with some fairly nice results.

library(tidyverse)

localdir <- '/home/jtimm/jt\_work/GitHub/data\_sets' ## devtools::install\_github("jaytimm/quicknews")

# Tweet-set

The tweet-set used here was accessed via the GWU Library, and subsequently “hydrated” using the Hydrator desktop application. Tweets were generated by members of the 116th House from 3 Jan 2019 to 7 May 2020. Subsequent analyses are based on a sample of 500 tweets/lawmaker containing shared URLs.

setwd(localdir)

house\_tweets <- readRDS('house116-sample-urls.rds') %>% filter(urls != '')

# Media bias data set

Media Bias/Fact Check is a fact-checking organization that classifies online news sources along two dimensions: (1) political bias and (2) factuality. These two scores (for ~850 sources) have been extracted by Baly et al. (2020).

setwd('/home/jtimm/jt\_work/GitHub/packages/quicknews/data-raw') ## emnlp18 <- read.csv('emnlp18-corpus.tsv', sep = '\t') acl2020 <- read.csv('acl2020-corpus.tsv', sep = '\t')

A sample of this data set is presented below.

set.seed(221) acl2020 %>%

group\_by(fact, bias) %>% sample\_n(1) %>%

# ungroup() %>%

select(source\_url\_normalized, fact, bias) %>% # spread(bias, source\_url\_normalized) %>% knitr::kable()

### source\_url\_normalized fact bias

wn.com high center

dailydot.com high left yellowhammernews.com high right freakoutnation.com low left

### source\_url\_normalized fact bias

christianaction.org low right

wionews.com mixed center

extranewsfeed.com mixed left

lifenews.com mixed right

# Resolving shortened URLs

The quicknews package is a collection of tools for navigating the online news landscape; here, we detail a simple workflow for researchers to use for multi-threaded URL un-shortening. As a three step process: (1) identify URLs that have been shortened via qnews\_clean\_urls, (2) split vector of URLs into multiple batches via qnews\_split\_batches for distribution across multiple cores, and (3) resolve shortened URLs via qnews\_unshorten\_urls.

## step 1

shortened\_urls <- quicknews::qnews\_clean\_urls(url = house\_tweets$urls)

%>%

filter(is\_short == 1)

## step 2

batch\_urls <- shortened\_urls %>% quicknews::qnews\_split\_batches(n = 12)

## step 3

unshortened\_urls <- parallel::mclapply(lapply(batch\_urls, "[[", 1),

quicknews::qnews\_unshorten\_urls, seconds = 10,

mc.cores = 12)

unshortened\_urls1 <- data.table::rbindlist(unshortened\_urls)

# Shared news media sources

Next, we update the original tweet-set with the resolved URLs from above; we also extract domain information from each shared link in our data set.

full\_tweets <- house\_tweets %>%

left\_join(unshortened\_urls1, by = c('urls' = 'short\_url')) %>% mutate(long\_url = ifelse(is.na(long\_url), urls, long\_url),

source = gsub('(http)(s)?(://)(www\\.)?', '', long\_url), source = gsub('/.\*$', '', source),

user\_screen\_name = toupper(user\_screen\_name)) ###

The list below details some less useful domains that we can remove from the data frame of shared URLs.

junks <- c('facebook', 'lnkd.in',

'twitter', 'youtube', 'youtu\\.be', 'instagram', 'twimg', 'tumblr',

'google', 'medium',

'vimeo', '\\.gov', 'actblue\\.com', 'bit\\.ly',

'ow\\.ly', 'timeout',

'myemail', 'apple.news', 'trib.al')

filt.tweets <- full\_tweets %>% filter(!grepl(paste0(junks, collapse = '|'), long\_url))

The table below summarizes some of the more frequently shared news media domains among lawmakers during the 116th congress. For good measure, domains are ranked by % coverage, which is the percentage of lawmakers that have shared a news link from a given domain in our data set. So, 94% (or 403/429) of House members shared content from The Hill, which compares to 49% for Fow News and only 15% for Breitbert.

share.summary <- filt.tweets %>% mutate(source = tolower(source)) %>% group\_by(source) %>%

summarize(n = n(), tweeters = length(unique(user\_screen\_name))) %>% ungroup() %>%

mutate(cover = round(tweeters/429\*100,1)) %>%

#left\_join(acl2020, by = c('source' = 'source\_url\_normalized')) %>% arrange(desc(tweeters)) %>%

filter(tweeters > 10)

**source n tweeters cover**

|  |  |  |  |
| --- | --- | --- | --- |
| thehill.com | 2977 | 403 | 93.9 |
| washingtonpost.com | 4853 | 384 | 89.5 |
| politico.com | 1782 | 354 | 82.5 |
| c-span.org | 1488 | 346 | 80.7 |
| nytimes.com | 4717 | 342 | 79.7 |
| cnn.com | 1802 | 323 | 75.3 |
| usatoday.com | 889 | 311 | 72.5 |
| cnbc.com | 973 | 309 | 72.0 |
| nbcnews.com | 1086 | 282 | 65.7 |
| wsj.com | 1043 | 277 | 64.6 |

# Media bias & tSNE

## Build matrix

To aggregate these data, we build a simple domain-lawmaker matrix, in which each domain/news organization is represented by the number of times each lawmaker has shared one of its news stories.

ft1 <- filt.tweets %>% group\_by(user\_screen\_name, source) %>% count() %>%

filter(source %in% share.summary$source) %>% tidytext::cast\_sparse(row = 'source',

column = 'user\_screen\_name', value = n)

ft2 <- as.matrix(ft1) #%>% Rtsne::normalize\_input()

Matrix top-left::

ft2[1:5, 1:5]

## AUSTINSCOTTGA08 BENNIEGTHOMPSON BETTYMCCOLLUM04 BILLPASCRELL

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ## abcnews.go.com |  |  |  | 1 |  | 4 |  | 0 |
| 3 |  |  |  |  |  |  |  |  |
| ## airforcetimes.com |  |  | 1 |  | 0 |  | 0 |  |
| 0 |  |  |  |  |  |  |  |  |
| ## ajc.com |  |  |  | 6 |  | 0 |  | 0 |
| 0 |  |  |  |  |  |  |  |  |
| ## bloomberg.com |  |  |  | 2 |  | 3 |  | 0 |
| 5 |  |  |  |  |  |  |  |  |
| ## c-span.org |  |  |  | 2 |  | 1 |  | 4 |
| 3 |  |  |  | | | | | |
| ## |  | BOBBYSCOTT |
| ## abcnews.go.com |  | 0 |
| ## airforcetimes.com |  | 0 |
| ## ajc.com |  | 0 |
| ## bloomberg.com |  | 2 |
| ## c-span.org  **TSNE** |  | 1 |
| set.seed(77) ## | 9 |  |

tsne <- Rtsne::Rtsne(X = ft2, check\_duplicates = FALSE)

tsne\_clean <- data.frame(descriptor\_name = rownames(ft1), tsne$Y) %>% #mutate(screen\_name = toupper(descriptor\_name)) %>% left\_join(acl2020, by = c('descriptor\_name' =

'source\_url\_normalized')) %>% replace(is.na(.), 'x')

## Plot

Per figure below, the first dimension of the tSNE plot does a fairly nice job capturing differences in **bias classifications** as presented by Media Bias/Fact Check, and results are generally intuitive. Factors underlying variation along the second dimension, however, are less clear, and do not appear to be capturing **factuality** in this case. *Note: news organizations indicated by orange Xs are not included in the MB/FC data set.*

split\_pal <- c('#3c811a',

'#395f81', '#9e5055', '#e37e00')

tsne\_clean %>% ggplot(aes(X1, X2)) + geom\_point(aes(col = bias,

shape = fact), size = 3) +

geom\_text(aes(label = descriptor\_name,

col = bias, shape = fact), #

size = 3,

check\_overlap = TRUE) + theme\_minimal() + theme(legend.position = "bottom") +

scale\_color\_manual(values = split\_pal) + xlab('Dimension 1') + ylab('Dimension 2')+ labs(title = "Measuring political bias")



## Bias score distributions

tsne\_clean %>% ggplot() +

geom\_density(aes(X1, fill = bias),

alpha = .4) + theme\_minimal() + theme(legend.position = "bottom") +

scale\_fill\_manual(values = split\_pal) +

ggtitle('Media bias scores by MB/FC bias classification')

