I decided to return to the analysis I conducted for the IRA tweets dataset. (You can read up on that analysis and R code below.)

files <- **c**("IRAhandle\_tweets\_1.csv",

"IRAhandle\_tweets\_2.csv",

"IRAhandle\_tweets\_3.csv",

"IRAhandle\_tweets\_4.csv",

"IRAhandle\_tweets\_5.csv",

"IRAhandle\_tweets\_6.csv",

"IRAhandle\_tweets\_7.csv",

"IRAhandle\_tweets\_8.csv",

"IRAhandle\_tweets\_9.csv")

my\_files <- **paste0**("~/Downloads/russian-troll-tweets-master/",files)

each\_file <- **function**(file) {

tweet <- **read\_csv**(file) }

**library**(tidyverse)

tweet\_data <- **NULL**

**for** (file **in** my\_files) {

temp <- **each\_file**(file)

temp$id <- **sub**(".csv", "", file)

tweet\_data <- **rbind**(tweet\_data, temp)

}

Note that this is a large file, with 2,973,371 observations of 16 variables. Let's do some cleaning of this dataset first. The researchers, Darren Linvill and Patrick Warren, identified 5 majors types of trolls:

* Right Troll: These Trump-supporting trolls voiced right-leaning, populist messages, but “rarely broadcast traditionally important Republican themes, such as taxes, abortion, and regulation, but often sent divisive messages about mainstream and moderate Republicans…They routinely denigrated the Democratic Party, e.g. @LeroyLovesUSA, January 20, 2017, “#ThanksObama We're FINALLY evicting Obama. Now Donald Trump will bring back jobs for the lazy ass Obamacare recipients,” the authors wrote.
* Left Troll: These trolls mainly supported Bernie Sanders, derided mainstream Democrats, and focused heavily on racial identity, in addition to sexual and religious identity. The tweets were “clearly trying to divide the Democratic Party and lower voter turnout,” the authors told FiveThirtyEight.
* News Feed: A bit more mysterious, news feed trolls mostly posed as local news aggregators who linked to legitimate news sources. Some, however, “tweeted about global issues, often with a pro-Russia perspective.”
* Hashtag Gamer: Gamer trolls used hashtag games—a popular call/response form of tweeting—to drum up interaction from other users. Some tweets were benign, but many “were overtly political, e.g. @LoraGreeen, July 11, 2015, “#WasteAMillionIn3Words Donate to #Hillary.”
* Fearmonger: These trolls, who were least prevalent in the dataset, spread completely fake news stories, for instance “that salmonella-contaminated turkeys were produced by Koch Foods, a U.S. poultry producer, near the 2015 Thanksgiving holiday.”

But a quick table of the results of the variable, account\_category, shows 8 in the dataset.

**table**(tweet\_data$account\_category)

##

## Commercial Fearmonger HashtagGamer LeftTroll NewsFeed

## 122582 11140 241827 427811 599294

## NonEnglish RightTroll Unknown

## 837725 719087 13905

The additional three are Commercial, Non-English, and Unknown. At the very least, we should drop the Non-English tweets, since those use Russian characters and any analysis I do will assume data are in English. I'm also going to keep only a few key variables. Then I'm going to clean up this dataset to remove links, because I don't need those for my analysis - I certainly wouldn't want to follow them to their destination. If I want to free up some memory, I can then remove the large dataset.

reduced <- tweet\_data %>%

**select**(author,content,publish\_date,account\_category) %>%

**filter**(account\_category != "NonEnglish")

**library**(qdapRegex)

*##*

*## Attaching package: 'qdapRegex'*

reduced$content <- **rm\_url**(reduced$content)

**rm**(tweet\_data)

Now we have a dataset of 2,135,646 observations of 4 variables. I'm planning on doing some analysis on my own of this dataset - and will of course share what I find - but for now, I thought I'd repeat a technique I've covered on this blog and demonstrate a new one.

**library**(tidytext)

tweetwords <- reduced %>%

**unnest\_tokens**(word, content) %>%

**anti\_join**(stop\_words)

*## Joining, by = "word"*

wordcounts <- tweetwords %>%

**count**(account\_category, word, sort = TRUE) %>%

**ungroup**()

**head**(wordcounts)

## # A tibble: 6 x 3

## account\_category word n

## <chr> <chr> <int>

## 1 NewsFeed news 124586

## 2 RightTroll trump 95794

## 3 RightTroll rt 86970

## 4 NewsFeed sports 47793

## 5 Commercial workout 42395

## 6 NewsFeed politics 38204

First, I'll conduct a TF-IDF analysis of the dataset. This code is a repeat from a [previous post](http://www.deeplytrivial.com/2018/07/statistics-sunday-more-text-analysis.html).

tweet\_tfidf <- wordcounts %>%

**bind\_tf\_idf**(word, account\_category, n) %>%

**arrange**(**desc**(tf\_idf))

tweet\_tfidf %>%

**mutate**(word = **factor**(word, levels = **rev**(**unique**(word)))) %>%

**group\_by**(account\_category) %>%

**top\_n**(15) %>%

**ungroup**() %>%

**ggplot**(**aes**(word, tf\_idf, fill = account\_category)) +

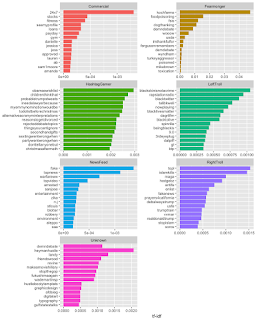
**geom\_col**(show.legend = FALSE) +

**labs**(x = **NULL**, y = "tf-idf") +

**facet\_wrap**(~account\_category, ncol = 2, scales = "free") +

**coord\_flip**()

*## Selecting by tf\_idf*

[](https://1.bp.blogspot.com/-SbdpFBt5MZk/W3Bd-Wvi3-I/AAAAAAAANaY/PBSzebwUl-cKMjQDOqHAKcovfHRpvdNlwCLcBGAs/s1600/tfidf_tweets1.png)

But another method of examining terms and topics in a set of documents is [Latent Dirichlet Allocation (LDA)](http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf), which can be conducted using the R package, topicmodels. The only issue is that LDA requires a [document term matrix](https://en.wikipedia.org/wiki/Document-term_matrix). But we can easily convert our wordcounts dataset into a DTM with the cast\_dtm function from tidytext. Then we run our LDA with topicmodels. Note that LDA is a random technique, so we set a random number seed, and we specify how many topics we want the LDA to extract (k). Since there are 6 account types (plus 1 unknown), I'm going to try having it extract 6 topics. We can see how well they line up with the account types.

tweets\_dtm <- wordcounts %>%

**cast\_dtm**(account\_category, word, n)

**library**(topicmodels)

tweets\_lda <- **LDA**(tweets\_dtm, k = 6, control = **list**(seed = 42))

tweet\_topics <- **tidy**(tweets\_lda, matrix = "beta")

Now we can pull out the top terms from this analysis, and plot them to see how they lined up.

top\_terms <- tweet\_topics %>%

**group\_by**(topic) %>%

**top\_n**(15, beta) %>%

**ungroup**() %>%

**arrange**(topic, -beta)

top\_terms %>%

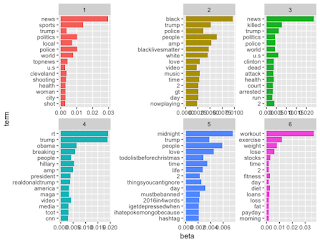
**mutate**(term = **reorder**(term, beta)) %>%

**ggplot**(**aes**(term, beta, fill = **factor**(topic))) +

**geom\_col**(show.legend = FALSE) +

**facet\_wrap**(~topic, scales = "free") +

**coord\_flip**()

[](https://4.bp.blogspot.com/-JAkFUPH8zic/W3BeKIg5ubI/AAAAAAAANac/ztZ8NY_jneUuH7zhoiCu49FdFqrKbZbLgCLcBGAs/s1600/LDA_tweets.png)

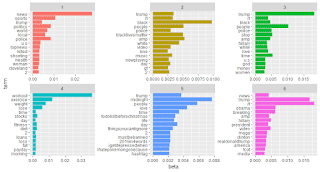
Based on these plots, I'd say the topics line up very well with the account categories, showing, in order: news feed, left troll, fear monger, right troll, hash gamer, and commercial. One interesting observation, though, is that Trump is a top term in 5 of the 6 topics.

Specifically, I returned to the LDA results, which looked like they lined up pretty well with the account categories identified by Darren Linvill and Patrick Warren. But with slightly altered code, we can confirm that or see if there’s more to the topics data than meets the eye. (Spoiler alert: There is more than meets the eye.)

I reran much of the original code – creating the file, removing non-English tweets and URLs, generating the DTM and conducting the 6-topic LDA.

I will note that the topics were numbered a bit differently than they were in my previous analysis. Here’s the new plot. The results look very similar to before. (LDA is a variational Bayesian method and there is an element of randomness to it, so the results aren’t a one-to-one match, but they’re very close.)

top\_terms <- tweet\_topics %>%  
 group\_by(topic) %>%  
 top\_n(15, beta) %>%  
 ungroup() %>%  
 arrange(topic, -beta)  
  
top\_terms %>%  
 mutate(term = reorder(term, beta)) %>%  
 ggplot(aes(term, beta, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free") +  
 coord\_flip()

[](https://i1.wp.com/4.bp.blogspot.com/-hJygd-E_nK8/W3dCDRiQJxI/AAAAAAAANb8/lgM_BRBobUIsT6FQnFbnQEJbpfmqQnSbwCLcBGAs/s1600/LDA_results1.png?ssl=1)

Before, when I generated a plot of the LDA results, I asked it to give me the top 15 terms by topic. I’ll use the same code, but instead have it give the top topic for each term.

word\_topic <- tweet\_topics %>%  
 group\_by(term) %>%  
 top\_n(1, beta) %>%  
 ungroup()

I can then match this dataset up to the original tweetwords dataset, to show which topic each word is most strongly associated with. Because the word variable is known by two different variable names in my datasets, I need to tell R how to match.

tweetwords <- tweetwords %>%  
 left\_join(word\_topic, by = c("word" = "term"))

Now we can generate a crosstable, which displays the matchup between LDA topic (1-6) and account category (Commercial, Fearmonger, Hashtag Gamer, Left Troll, News Feed, Right Troll, and Unknown).

cat\_by\_topic <- table(tweetwords$account\_category, tweetwords$topic)  
cat\_by\_topic

##   
## 1 2 3 4 5 6  
## Commercial 38082 34181 49625 952309 57744 19380  
## Fearmonger 9187 3779 37326 1515 8321 4864  
## HashtagGamer 117517 103628 183204 31739 669976 81803  
## LeftTroll 497796 1106698 647045 94485 395972 348725  
## NewsFeed 2715106 331987 525710 91164 352709 428937  
## RightTroll 910965 498983 1147854 113829 534146 2420880  
## Unknown 7622 5198 12808 1497 11282 4605

This table is a bit hard to read, because it’s frequencies, and the total number of words for each topic and account category differ. But we can solve that problem by asking instead for proportions. I’ll have it generate proportions by column, so we can see the top account category associated with each topic.

options(scipen = 999)  
prop.table(cat\_by\_topic, 2) #column percentages - which topic is each category most associated with

##   
## 1 2 3 4 5  
## Commercial 0.008863958 0.016398059 0.019060352 0.740210550 0.028443218  
## Fearmonger 0.002138364 0.001812945 0.014336458 0.001177579 0.004098712  
## HashtagGamer 0.027353230 0.049714697 0.070366404 0.024670084 0.330013053  
## LeftTroll 0.115866885 0.530929442 0.248522031 0.073441282 0.195045686  
## NewsFeed 0.631967460 0.159268087 0.201918749 0.070859936 0.173735438  
## RightTroll 0.212036008 0.239383071 0.440876611 0.088476982 0.263106667  
## Unknown 0.001774095 0.002493699 0.004919395 0.001163588 0.005557225  
##   
## 6  
## Commercial 0.005856411  
## Fearmonger 0.001469844  
## HashtagGamer 0.024719917  
## LeftTroll 0.105380646  
## NewsFeed 0.129619781  
## RightTroll 0.731561824  
## Unknown 0.001391578

Category 1 is News Feed, Category 2 Left Troll, Category 4 Commercial, and Category 5 Hashtag Gamer. But look at Categories 3 and 6. For both, the highest percentage is Right Troll. Fearmonger is not most strongly associated with any specific topic. What happens if we instead ask for a proportion table by row, which tells us which category each topic most associated with?

prop.table(cat\_by\_topic, 1) #row percentages - which category is each topic most associated with

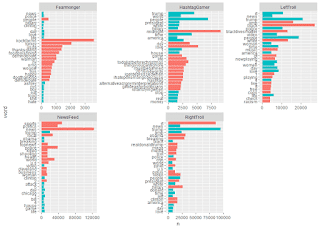
##   
## 1 2 3 4 5  
## Commercial 0.03307679 0.02968851 0.04310266 0.82714465 0.05015456  
## Fearmonger 0.14135586 0.05814562 0.57431684 0.02331056 0.12803114  
## HashtagGamer 0.09893111 0.08723872 0.15422939 0.02671932 0.56401601  
## LeftTroll 0.16106145 0.35807114 0.20935083 0.03057054 0.12811638  
## NewsFeed 0.61073827 0.07467744 0.11825366 0.02050651 0.07933866  
## RightTroll 0.16190164 0.08868197 0.20400284 0.02023031 0.09493132  
## Unknown 0.17720636 0.12085000 0.29777736 0.03480424 0.26229889  
##   
## 6  
## Commercial 0.01683284  
## Fearmonger 0.07483998  
## HashtagGamer 0.06886545  
## LeftTroll 0.11282966  
## NewsFeed 0.09648546  
## RightTroll 0.43025192  
## Unknown 0.10706315

Based on these results, Fearmonger now seems closest to Category 3 and Right Troll with Category 6. But Right Troll also shows up on Categories 3 (20%) and 1 (16%). Left Trolls show up in these categories at almost each proportions. It appears, then, that political trolls show strong similarity in topics with Fearmongers (stirring things up) and News Feed (“informing”) trolls. Unknown isn’t the top contributer to any topic, but it aligns with Category 3 (showing elements of Fearmongering) and 5 (showing elements of Hashtag Gaming). Let’s focus in on 5 categories.

categories <- c("Fearmonger", "HashtagGamer", "LeftTroll", "NewsFeed", "RightTroll")  
  
politics\_fear\_hash <- tweetwords %>%  
 filter(account\_category %in% categories)  
  
PFH\_counts <- politics\_fear\_hash %>%  
 count(account\_category, topic, word, sort = TRUE) %>%  
 ungroup()

For now, let’s define our topics like this: 1 = News Feed, 2 = Left Troll, 3 = Fearmonger, 4 = Commercial, 5 = Hashtag Gamer, and 6 = Right Troll. We’ll ask R to go through our PFH dataset and tell us when account category topic matches and when it mismatches. Then we can look at these terms.

PFH\_counts$match <- ifelse(PFH\_counts$account\_category == "NewsFeed" & PFH\_counts$topic == 1,PFH\_counts$match <- "Match",  
 ifelse(PFH\_counts$account\_category == "LeftTroll" & PFH\_counts$topic == 2,PFH\_counts$match <- "Match",  
 ifelse(PFH\_counts$account\_category == "Fearmonger" & PFH\_counts$topic == 3,PFH\_counts$match <- "Match",  
 ifelse(PFH\_counts$account\_category == "HashtagGamer" & PFH\_counts$topic == 5,PFH\_counts$match <- "Match",  
 ifelse(PFH\_counts$account\_category == "RightTroll" & PFH\_counts$topic == 6,PFH\_counts$match <- "Match",  
 PFH\_counts$match <- "NonMatch")))))  
  
top\_PFH <- PFH\_counts %>%  
 group\_by(account\_category, match) %>%  
 top\_n(15, n) %>%  
 ungroup() %>%  
 arrange(account\_category, -n)  
  
top\_PFH %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word, n, fill = factor(match))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~account\_category, scales = "free") +  
 coord\_flip()

[](https://i1.wp.com/1.bp.blogspot.com/-e2VLYBV1NCw/W3dCcoOra-I/AAAAAAAANcE/X-BLMrZlIYIA735_zG4-2gJI2JXKox_iwCLcBGAs/s1600/match_mismatch.png?ssl=1)

Red indicates a match and blue indicates a mismatch. So when Fearmongers talk about food poisoning or Koch Farms, it’s a match, but when they talk about Hillary Clinton or the police, it’s a mismatch. Terms like “MAGA” and “CNN” are matches for Right Trolls but “news” and “love” are mismatches. Left Trolls show a match when tweeting about “Black Lives Matter” or “police” but a mismatch when tweeting about “Trump” or “America.” An interesting observation is that Trump is a mismatch for every topic it’s displayed under on the plot. (Now, realdonaldtrump, Trump’s Twitter handle, is a match for Right Trolls.) So where does that term, and associated terms like “Donald” belong?

tweetwords %>%  
 filter(word %in% c("donald", "trump"))

## # A tibble: 157,844 x 7  
## author publish\_date account\_category id word topic beta  
##   
## 1 10\_GOP 10/1/2017 22:43 RightTroll C:/Users/~ trump 3 0.0183   
## 2 10\_GOP 10/1/2017 23:52 RightTroll C:/Users/~ trump 3 0.0183   
## 3 10\_GOP 10/1/2017 2:47 RightTroll C:/Users/~ dona~ 3 0.00236  
## 4 10\_GOP 10/1/2017 2:47 RightTroll C:/Users/~ trump 3 0.0183   
## 5 10\_GOP 10/1/2017 3:47 RightTroll C:/Users/~ trump 3 0.0183   
## 6 10\_GOP 10/10/2017 20:57 RightTroll C:/Users/~ trump 3 0.0183   
## 7 10\_GOP 10/10/2017 23:42 RightTroll C:/Users/~ trump 3 0.0183   
## 8 10\_GOP 10/11/2017 22:14 RightTroll C:/Users/~ trump 3 0.0183   
## 9 10\_GOP 10/11/2017 22:20 RightTroll C:/Users/~ trump 3 0.0183   
## 10 10\_GOP 10/12/2017 0:38 RightTroll C:/Users/~ trump 3 0.0183   
## # ... with 157,834 more rows

These terms apparently were sorted into Category 3, which we’ve called Fearmongers. Once again, this highlights the similarity between political trolls and fearmongering trolls in this dataset.