Classifying News Sources

Frances Hung

10/21/2017

Intro

"Fake news" has been a concern for both sides of the political spectrum. We can look at topic coverage over different news sources by analyzing the topics, or keywords, which appear frequently across all news headlines during a given day. Are there keyword frequency patterns (perhaps we can call them "fingerprints") which can be used to identify political leanings or biasedness among news sources?

In the first part of this project, I attempted to visualize these fingerprints. With the help of NewsAPI (https://newsapi.org/) and some regex, I cleaned the headlines of a few news sources and found keywords. I then plotted the frequencies of these keywords in a bubble chart.

```
require(httr)
require(jsonlite)
require(lubridate)
require(dplyr)
require(data.table)
require(tidyr)
require(ggplot2)
require(tm)
require(stringr)
```

```
make_dataframe <- function(url) {
    headlines <- GET(url)$content %>% rawToChar() %>% fromJSON() %>% .[3] %>%
        as.data.frame() %>% .$articles.title %>% as.data.frame()
    colnames(headlines) <- GET(url)$content %>% rawToChar() %>% fromJSON() %>%
        .[3] %>% as.data.frame() %>% .$articles.source %>% .$id %>% .[1]
    return(headlines)
}

# clean headlines by removing punctuation
remove_stops <- function(headlines) {
    heads = as.vector(headlines) %>% tolower() %>% gsub("'s|[[:punct:]]", "",
        .) %>% gsub("[[:space:]]", " ", .)
    stop_regex = paste(stopwords("en"), collapse = "\\b|\\b")
    stop_regex = paste0("\\b", stop_regex, "\\b")
    heads = stringr::str_replace_all(heads, stop_regex, "")
    return(heads)
}
```

Reading in Headlines

```
al_jazeera <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=al-jazeera-english&apiKey=9b
# colnames(al_jazeera)[2]<-'al_jazeera'
bbc <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=bbc-news&apiKey=9bfcf9f72ada452aa95
breitbart <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=breitbart-news&apiKey=9bfcf9f
reuters <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=reuters&apiKey=9bfcf9f72ada452a
times <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=time&apiKey=9bfcf9f72ada452aa953c
google_news <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=google-news&apiKey=9bfcf9f7
cbs <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=cbs-news&apiKey=9bfcf9f72ada452aa95
ap <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=associated-press&apiKey=9bfcf9f72ada
fox <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=fox-news&apiKey=9bfcf9f72ada452aa95
huff_post <- make_dataframe("https://newsapi.org/v2/top-headlines?sources=the-huffington-post&apiKey=9b
```

```
headlines.short <- unlist(lapply(c(reuters, bbc, google_news, fox, breitbart,
    huff_post), remove_stops))

# dataframe of words repeated more than twice across headlines
headlines.short <- unlist(strsplit(headlines.short, " ")) %>% table() %>% .[(.) >
    2] %>% as.data.frame() %>% .[-1, ]
colnames(headlines.short) <- c("word", "freq")
```

What proportion of headlines in one source contain a certain key word(s)?

```
topic_contain <- function(list, dataset) {
    p <- TRUE
    for (i in length(list)) {
        new <- grepl(list[i], dataset[, 1])
        p <- (p & new)
    }
    sum(p)/10
}
```

Applying the above function to multiple key words

```
topics <- function(dataset, terms) {
    as.data.frame(unlist(lapply(terms, topic_contain, dataset))) %>% unlist()
}
```

Visualizing Keyword Frequencies in New Sources

I've taken the words which appear more than twice among headlines of six news sources: Reuters, BBC, Google News, Fox, Breitbart, and the Huffington Post.

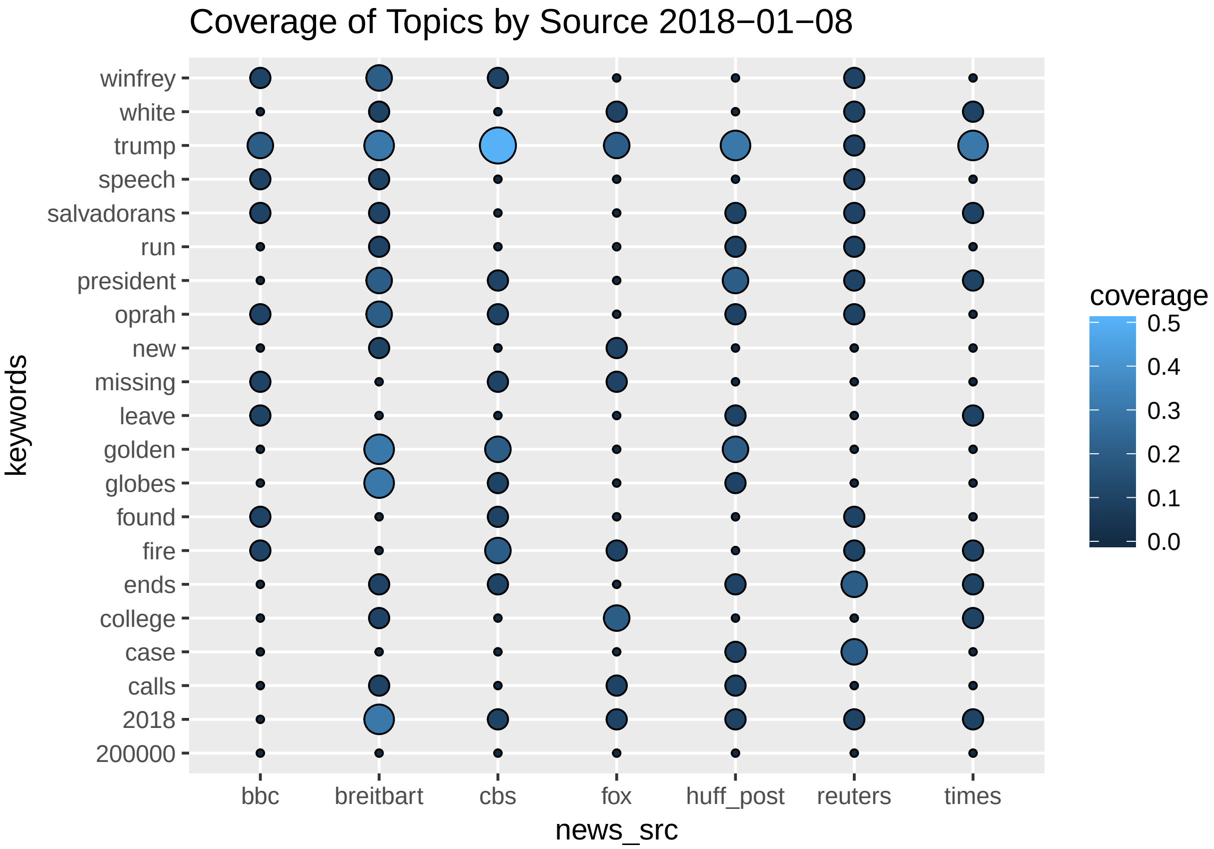
I then plotted frequencies of each of these words for each of 7 news sources on a bubble plot. The size and color of each bubble represent the frequency of a word in a given news source.

```
# function which takes a news source and returns the frequency of keywords
# appearing in that source's headlines requires final.keywords$words, a
# column of keywords
final_col <- function(news_source) {
    news <- sapply(news_source, tolower)
    single <- topics(news, headlines.short$word) %>% as.data.frame()
    rownames(single) <- headlines.short$word %>% as.character()
    return(single)
}

# function which applies the above function to many news sources and
# aggregates them as a dataframe
freq.keywords <- function(news) {
    final_table <- as.data.frame(sapply(news, function(x) final_col(x)))
    rownames(final_table) <- headlines.short$word
    return(final_table)
}

final_table <- freq.keywords(list(reuters, bbc, cbs, breitbart, huff_post, fox,
    times))
colnames(final_table) <- c("reuters", "bbc", "cbs", "breitbart", "huff_post",
    "fox", "times")
final_table <- final_table %>% setDT(keep.rownames = TRUE)

final_table %>% gather(news_src, coverage, -rn) %>% ggplot(aes(x = news_src,
    y = rn)) + geom_point(aes(size = coverage, fill = coverage), shape = 21) +
    guides(size = FALSE) + ylab("keywords") + ggtitle(paste("Coverage of Topics by Source",
    Sys.Date()))
```



Future Steps

I'm planning on storing key words and frequencies for each news source in an SQL database, then using neural nets (likely via TensorFlow) to attempt to classify keyword patterns into conservative vs. liberal, extreme vs. moderate, etc.