

Blogbeitrag Seminar WebDevelopment 2021

Gina Messerli

28 05 2021

```
#Datenvorbereitung
#WD setzen
setwd("~/Master UZH/Polito/Vorlesungen/FS21_Datenjournalismus/Daten/Facebook")

#Load libraries
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
library(quanteda)

## Warning: package 'quanteda' was built under R version 4.0.4
## Package version: 2.1.2
## Parallel computing: 2 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
##
## Attaching package: 'quanteda'
## The following object is masked from 'package:utils':
##
##   View
library(readtext)

## Warning: package 'readtext' was built under R version 4.0.4
library(syuzhet)

## Warning: package 'syuzhet' was built under R version 4.0.4
library(cld3)

## Warning: package 'cld3' was built under R version 4.0.4
library(ggplot2)
library(hrbrthemes)
```

```

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
library(bbplot)
library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.0.4
## Loading required package: RColorBrewer
library(wordcloud2)

## Warning: package 'wordcloud2' was built under R version 4.0.4
library(RColorBrewer)
library(quanteda.dictionaries)
library(tidyr)
library(seededlda)

## Warning: package 'seededlda' was built under R version 4.0.5
##
## Attaching package: 'seededlda'
## The following object is masked from 'package:stats':
##
##   terms
library(data.table)

##
## Attaching package: 'data.table'
## The following objects are masked from 'package:dplyr':
##
##   between, first, last
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v tibble  3.0.1    v stringr 1.4.0
## v readr   1.4.0    v forcats 0.5.0
## v purrr   0.3.4
## Warning: package 'readr' was built under R version 4.0.5
## -- Conflicts ----- tidyverse_conflicts() --
## x data.table::between() masks dplyr::between()
## x dplyr::filter()       masks stats::filter()
## x data.table::first()   masks dplyr::first()
## x dplyr::lag()          masks stats::lag()
## x data.table::last()    masks dplyr::last()
## x purrr::transpose()    masks data.table::transpose()
#FB-Daten einlesen
fb_parties <- readRDS("parties_on_facebook.rds")

#Filtern: Nur Posts ab 02.03. miteinbeziehen
filtered_fb_parties_2020 <- subset(fb_parties, Datum >= "2020-03-02")

```

```

#Parteien recordieren
filtered_fb_parties_2020 <- filtered_fb_parties_2020 %>% dplyr::mutate(party_new=case_when(Account_name
  Account_name %in% c("FDP", "PLR") ~ "FDP",
  Account_name %in% c("Grüne Schweiz","Les Verts suisses") ~ "Grüne",
  Account_name %in% c("Grünliberale","Vert'libéraux") ~ "Grünliberale",
  Account_name %in% "Piratenpartei Schweiz" ~ "Piratenpartei",
  Account_name %in% c("SP Schweiz", "PS Suisse
") ~ "SP",
  Account_name %in% c("SVP Schweiz","UDC Suisse
") ~ "SVP",
  Account_name %in% "up schweiz" ~ "up"
))

#Up-Partei & Piratenpartei rauslöschen
filtered_fb_parties_2020 <- filtered_fb_parties_2020 %>% dplyr::filter(party_new!="up"&party_new!="Pira

#####Corpus mit allen Posts#####
#Corpus machen
corpus_fb1 <- corpus(filtered_fb_parties_2020, text_field="Text")

#Dem Korpus eine bedeutungsvolle ID hinzufügen
docid <- paste(filtered_fb_parties_2020$Datum,
  filtered_fb_parties_2020$party_new,
  filtered_fb_parties_2020$cutoff,
  sep=" ")
docnames(corpus_fb1) <- docid

#Aus Korpus Tokens machen und dabei Zahlen, Satzzeichen, Symbole und URLs entfernen
token_fb_all <- tokens(corpus_fb1, "word", remove_number=T, remove_url = T, remove_punct = T, remove_sy

#Stopworte entfernen mit der Default-Funktion von Quanteda
token_fb_all <- tokens_select(token_fb_all, pattern = stopwords("de"), selection = "remove")
token_fb_all <- tokens_select(token_fb_all, pattern = stopwords("fr"), selection = "remove")
token_fb_all <- tokens_select(token_fb_all, pattern = stopwords("it"), selection = "remove")

#Custom Stopworte rauslöschen
token_fb_all <- tokens_select(token_fb_all, pattern = c("dass", "innen", "liegt", "obwohl", "lag", "wei

#Tokens stemmen in allen Sprachen
token_fb_all <- tokens_wordstem(token_fb_all, language = "de")
token_fb_all <- tokens_wordstem(token_fb_all, language = "fr")
token_fb_all <- tokens_wordstem(token_fb_all, language = "it")

#Document-Feature Matrix machen; alles kleinschreiben

```



```

dfm_corona_de <- dfm(token_fb_de_corona)

#Piechart & Barchart
#####Piechart Grafik#####
#Dictionary matches: wie oft wurde insgesamt über Corona gesprochen
dfm_corona2 <- tokens_lookup(token_fb_all, dict_corona, nomatch="nicht corona") %>% dfm()
features_dfm_corona2 <- textstat_frequency(dfm_corona2)

#Berechnen: Erwähnung von Corona insgesamt über alle Parteien
freq_akteure_corona2 <- features_dfm_corona2 %>%
  mutate(doctot = sum(docfreq)) %>% #doctot: Alle Posts zu einem Datum (über Bundesrat und nicht über B
  mutate(prozent = 100 * docfreq/doctot) #Vorbereitung: Wieviele Posts zu einem Datum sind über den Bun

#Vorbereitung für Piechart
freq_akteure_corona2$prozent <- round(freq_akteure_corona2$prozent, digits=2)
freq_akteure_corona2$ymax = cumsum(freq_akteure_corona2$prozent)
freq_akteure_corona2$ymin = c(0, head(freq_akteure_corona2$ymax, n=-1))
freq_akteure_corona2$labelPosition <- (freq_akteure_corona2$ymax + freq_akteure_corona2$ymin) / 2

#Labels für Piechart
freq_akteure_corona2$label <- paste0(freq_akteure_corona2$feature, "\n Wert: ", freq_akteure_corona2$pr

#Piechart plotten
piechart <- ggplot(freq_akteure_corona2, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=feature)) +
  geom_rect() +
  geom_text(x=5.2, aes(y=labelPosition, label=label, color=feature), size=5) + # x here controls label
  scale_fill_manual(values=c("black", "#7E7E7E")) +
  scale_color_manual(values=c("black", "#7E7E7E")) +
  coord_polar(theta="y") +
  xlim(c(1, 6)) +
  theme_void() +
  theme(legend.position = "none") +
  geom_text(y = 1, x = 1, label = "n=4055", size=5)

#####Barplot: Welche Partei hat das Thema wie oft aufgegriffen#####
#Wie oft hat welche Partei über Corona gesprochen? Corona-Dictionary matches
dfm_corona <- tokens_lookup(token_fb_all, dict_corona, nomatch="_unmatched") %>% dfm()

#Matched: Welche Partei hat wie häufig über Corona gesprochen?
features_dfm_corona <- textstat_frequency(dfm_corona, group = "party_new")

#Berechnen: Welche Partei hat wie häufig über Corona gesprochen?
freq_akteure_corona <- features_dfm_corona %>% group_by(group) %>% #group=Datum
  mutate(doctot = sum(docfreq)) %>% #doctot: Alle Posts zu einem Datum (über Bundesrat und nicht über B
  mutate(prozent = 100 * docfreq/doctot) %>% #Vorbereitung: Wieviele Posts zu einem Datum sind über den
  filter(feature!="_unmatched")

```



```

keyness_green_data$party <- "Grüne"

#Keyness Grünliberale
keyness_greenlib_plot <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Grünliberale")

#Zu Dataframe machen und ID Variable hinzufügen
keyness_greenlib_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Grünliberale")
keyness_greenlib_data$party <- "Grünliberale"

#Keyness FDP
keyness_fdp_plot <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("FDP") %>% textp

#Zu Dataframe machen und ID variable hinzufügen
keyness_fdp_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("FDP") %>% head
keyness_fdp_data$party <- "FDP"

#Keyness Mitte
keyness_mitte_plot <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Die Mitte") %>% head

#Zu dataframe machen und ID Variable hinzufügen
keyness_mitte_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Die Mitte") %>% head
keyness_mitte_data$party <- "Mitte"

#Alle Keyness-dataframes mergen
keyness_all <- rbind(keyness_svp_data, keyness_sp_data, keyness_green_data, keyness_greenlib_data, keyness_mitte_data)

#Als CSV exportieren
write.csv(keyness_all, "keyness_all.csv", row.names = T)

#Barplot: Keyness nach Partei

#SVP; ID-Variable hinzufügen
keyness_svp_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("SVP") %>% head
keyness_svp_data$party <- "SVP"

#SP; ID-Variable hinzufügen
keyness_sp_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("SP") %>% head(2)
keyness_sp_data$party <- "SP"

#Grüne; ID-Variable hinzufügen
keyness_green_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Grüne") %>% head
keyness_green_data$party <- "Grüne"

#Grünliberale; ID-Variable hinzufügen
keyness_greenlib_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Grünliberale")
keyness_greenlib_data$party <- "Grünliberale"

#FDP; ID-Variable hinzufügen
keyness_fdp_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("FDP") %>% head
keyness_fdp_data$party <- "FDP"

```

```
#Mitte; ID-Variable hinzufügen
keyness_mitte_data <- dfm_group(dfm_corona_de, groups = "party_new") %>% textstat_keyness("Die Mitte")
keyness_mitte_data$party <- "Mitte"

#Alle Keynes-dataframes mergen
keyness_all <- rbind(keyness_svp_data, keyness_sp_data, keyness_green_data, keyness_greenlib_data, keyness_

#Als CSV exportieren, um es dann in D3 verwenden zu können
write.csv(keyness_all, "keyness_all.csv", row.names = T)
```