Dictionary Analysis

Riccardo Ruta

5/2022

Contents

Dictionary analysis

At the level of political parties, which ones make most use of populist rhetoric?

At the level of individual politicians, which ones make most use of populist rhetoric?

I use 3 dictionary to perform the analysis

- Rooduijn & Pauwels: Rooduijn, M., and T. Pauwels. 2011. "Measuring Populism: Comparing Two Methods of Content Analysis." West European Politics 34 (6): 1272–1283.
- Decadri & Boussalis: Decadri, S., & Boussalis, C. (2020). Populism, party membership, and language complexity in the Italian chamber of deputies. Journal of Elections, Public Opinion and Parties, 30(4), 484-503.
- Grundl: Gründl J. Populist ideas on social media: A dictionary-based measurement of populist communication. New Media & Society. December 2020.
- Decadri & Boussalis + Grundl: this is simply a more extended version of the D&B dictionary, which also contains some terms taken from Grundl.

Create the dictionary

I imported the excel file with the words for the dictionaries, excluding NA's.

```
# import dictionaries file
dict <-  read_excel("data/populism_dictionaries.xlsx")
variable.names(dict)
```

```
## [1] "Rooduijn_Pauwels_Italian"
## [2] "Grundl_Italian_adapted"
## [3] "Decadri_Boussalis"
## [4] "Decadri_Boussalis_Grundl_People"
## [5] "Decadri_Boussalis_Grundl_Common Will"
## [6] "Decadri_Boussalis_Grundl_Elite"
```

```
# create the dictionary
Rooduijn_Pauwels_Italian <-
  dictionary(list(populism =
                    (dict$Rooduijn_Pauwels_Italian
                     [!is.na(dict$Rooduijn_Pauwels_Italian)])))

Grundl_Italian_adapted <-
  dictionary(list(populism =
                    dict$Grundl_Italian_adapted
                  [!is.na(dict$Grundl_Italian_adapted)]))

Decadri_Boussalis <-
  dictionary(list(populism =
                    dict$Decadri_Boussalis
                  [!is.na(dict$Decadri_Boussalis)]))

Decadri_Boussalis_Grundl <-
  dictionary(list(people =
                    dict$Decadri_Boussalis_Grundl_People
                  [!is.na(dict$Decadri_Boussalis_Grundl_People)],
                  common_will =
                    dict$`Decadri_Boussalis_Grundl_Common Will`
                  [!is.na(dict$`Decadri_Boussalis_Grundl_Common Will`)],
                  elite =
                    dict$Decadri_Boussalis_Grundl_Elite
                  [!is.na(dict$Decadri_Boussalis_Grundl_Elite)]))
```

```
dictionaries <- c("Rooduijn_Pauwels_Italian", "Grundl_Italian_adapted",
                  "Decadri_Boussalis","Decadri_Boussalis_Grundl")
n.words <- c(length(Rooduijn_Pauwels_Italian$populism),
           length(Grundl_Italian_adapted$populism),
           length(Decadri_Boussalis$populism),
           (length(Decadri_Boussalis_Grundl$people)+
              length(Decadri_Boussalis_Grundl$common_will)+
              length(Decadri_Boussalis_Grundl$elite))
            )
number_of_words <- data.frame(dictionaries,n.words)
kable(number_of_words)
```

dictionaries	n.words
Rooduijn_Pauwels_Italian	18
Grundl_Italian_adapted	135
Decadri_Boussalis	25
Decadri_Boussalis_Grundl	77

Group and weight the dfm

```
# By party & month
dfm_weigh_p_month <- dfm_group(DFM, groups = interaction(party_id, month)) %>% dfm_weight(scheme = "prop
# By party & quarter
dfm_weigh_p_quart <- dfm_group(DFM, groups = interaction(party_id, quarter)) %>% dfm_weight(scheme = "pr
# By party & year
dfm_weigh_p_year <- dfm_group(DFM, groups = interaction(party_id, year)) %>% dfm_weight(scheme = "prop"
# By party & day
dfm_weigh_p_day <- dfm_group(DFM, groups = interaction(party_id, date)) %>% dfm_weight(scheme = "prop")
```

Apply dictionary

Decadri_Boussalis_Grundl

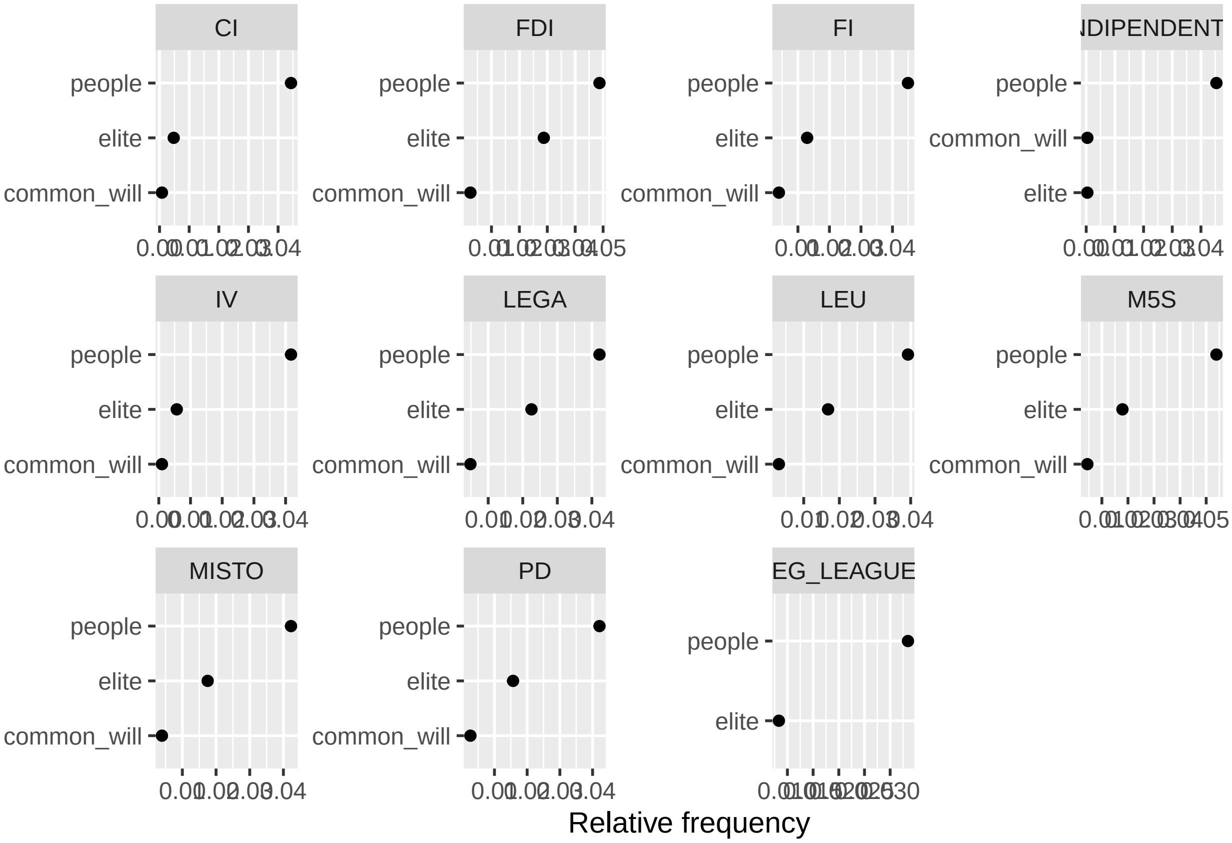
Level of sparsity

daily: 12.08%

weekly: 0.55%

monthly: 0%

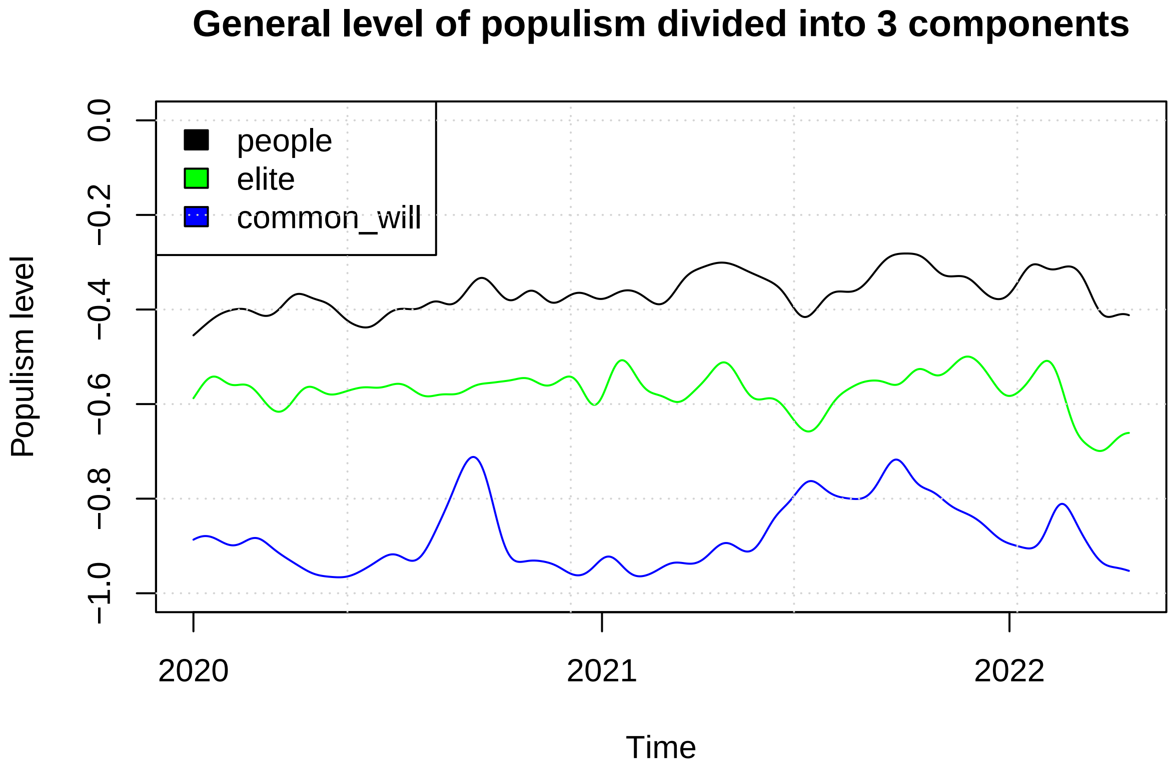
```
# Dictionary analysis with Decadri_Boussalis_Grundl
# By quarter
dfm_dict1  <- dfm_lookup(dfm_weigh_p_quart, dictionary = Decadri_Boussalis_Grundl)
# By date
dfm_by_date1 <- dfm_lookup(dfm_weigh_p_day, dictionary = Decadri_Boussalis_Grundl)
```



Looking at the populist rhetoric for each party divided into the 3 components people-centrism, anti-elitism and common-will, we note that the most frequent components is People-centrism.

General level of populism in time divided into 3 components

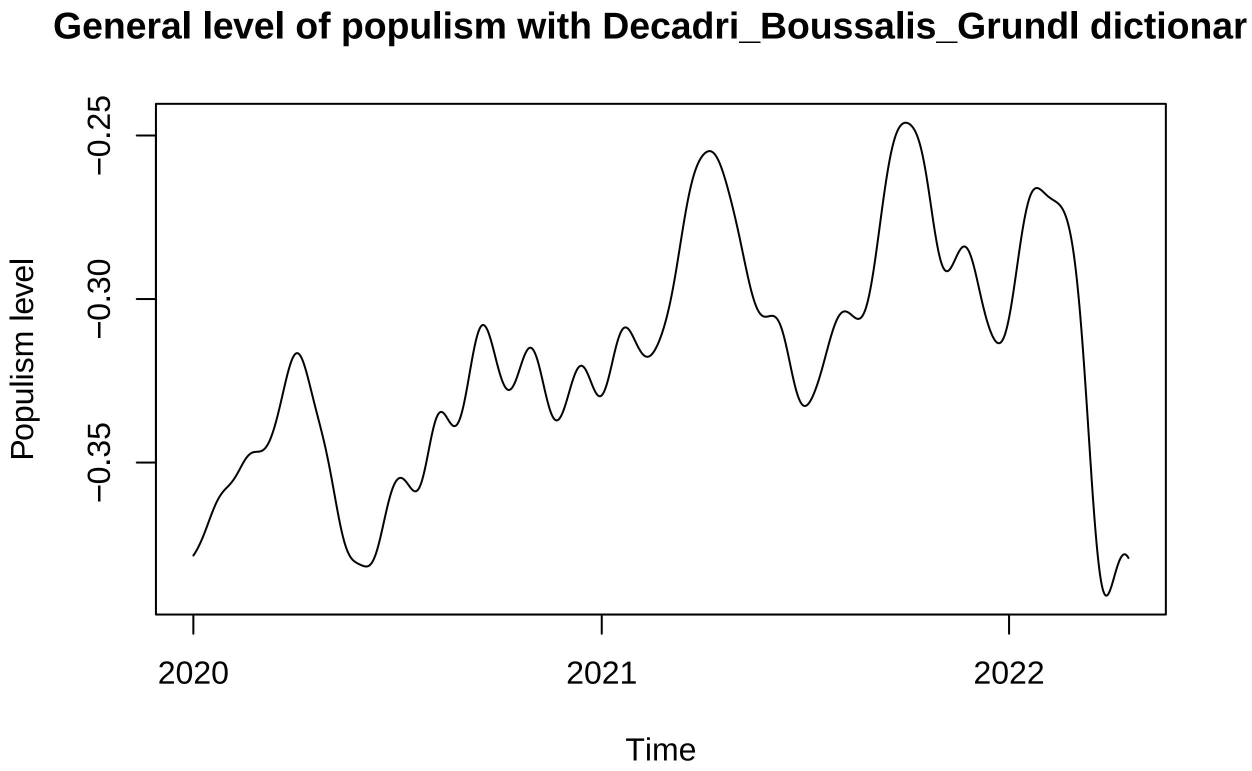
```r
dat_smooth1.1 <- ksmooth(x = dfm_by_date1$date,
                      y = dfm_by_date1[,"people"] - !dfm_by_date1[,"people"],
                      kernel = "normal", bandwidth = 30)

dat_smooth1.2 <- ksmooth(x = dfm_by_date1$date,
                      y = dfm_by_date1[,"common_will"] - !dfm_by_date1[,"common_will"],
                      kernel = "normal", bandwidth = 30)

dat_smooth1.3 <- ksmooth(x = dfm_by_date1$date,
                      y = dfm_by_date1[,"elite"] - !dfm_by_date1[,"elite"],
                      kernel = "normal", bandwidth = 30)

plot_time_1 <- plot(dat_smooth1.1$x, dat_smooth1.1$y,
                    type = "l", ylab = "Populism level", xlab = "Time",  ylim = c(-1,0), xlim = c(min(d

  lines(dat_smooth1.2$x, dat_smooth1.2$y,
        type = "l", ylab = "Populism level", xlab = "Time",col = "blue")

  lines(dat_smooth1.3$x, dat_smooth1.3$y,
        type = "l", ylab = "Populism level", xlab = "Time", col = "green")

  legend("topleft", legend = c("people","elite","common_will"),
         fill = c("black", "green", "blue"))
  title(main = "General level of populism divided into 3 components")
  grid()
```



This plot is coherent with the previous one and show us that people is the Component that score better.

General level of populism in time

```
dat_smooth1 <- ksmooth(x = dfm_by_date1$date,
                       y = ((dfm_by_date1[,"people"]+
                              dfm_by_date1[,"common_will"]+
                              dfm_by_date1[,"elite"])/3) -
                         !((dfm_by_date1[,"people"]+
                              dfm_by_date1[,"common_will"]+
                              dfm_by_date1[,"elite"])/3),
                       kernel = "normal", bandwidth = 30)

plot_time_1 <- plot(dat_smooth1$x, dat_smooth1$y,
                    type = "l", ylab = "Populism level", xlab = "Time")
title(main = "General level of populism with Decadri_Boussalis_Grundl dictionary")
```



### Most populist parliamentary group for each component

```
# Most populist parliamentary group
dfm_dict1_tstat_party <- textstat_frequency(dfm_dict1, groups = party_id)
kable(dfm_dict1_tstat_party %>% filter(feature == "people") %>% slice_max(frequency, n = 20))
```

	feature	frequency	rank	docfreq	group
22	people	0.0539309	1	10	M5S
4	people	0.0486981	1	10	FDI
10	people	0.0454089	1	10	INDIPENDENTE
7	people	0.0449102	1	10	FI
1	people	0.0443528	1	10	CI
25	people	0.0422541	1	10	MISTO
16	people	0.0422111	1	10	LEGA
28	people	0.0421103	1	10	PD
13	people	0.0417078	1	10	IV
19	people	0.0392343	1	10	LEU
31	people	0.0334640	1	9	REG_LEAGUES

```r
kable(dfm_dict1_tstat_party %>% filter(feature == "elite") %>% slice_max(frequency, n = 20))
```

	feature	frequency	rank	docfreq	group
5	elite	0.0287440	2	10	FDI
17	elite	0.0225301	2	10	LEGA
23	elite	0.0178936	2	10	M5S
26	elite	0.0175129	2	10	MISTO
20	elite	0.0168347	2	10	LEU
29	elite	0.0156834	2	10	PD
8	elite	0.0129287	2	10	FI
32	elite	0.0083365	2	10	REG_LEAGUES
14	elite	0.0056664	2	9	IV
2	elite	0.0047789	2	9	CI
12	elite	0.0004012	2	2	INDIPENDENTE

```r
kable(dfm_dict1_tstat_party %>% filter(feature == "common_will") %>% slice_max(frequency, n = 20))
```

	feature	frequency	rank	docfreq	group
18	common_will	0.0048497	3	10	LEGA
24	common_will	0.0044336	3	10	M5S
9	common_will	0.0039613	3	10	FI
27	common_will	0.0039426	3	9	MISTO
21	common_will	0.0030362	3	9	LEU
30	common_will	0.0026507	3	10	PD
6	common_will	0.0024601	3	10	FDI
15	common_will	0.0009923	3	4	IV
3	common_will	0.0008098	3	4	CI
11	common_will	0.0004012	2	2	INDIPENDENTE

Most populist parliamentary group

```r
# Add variable for the general level of populism
dfm_dict1_tstat_party <- dfm_dict1_tstat_party %>%
  group_by(group) %>% mutate(populism = mean(frequency))
dfm_dict1_tstat_party_filtered <- dfm_dict1_tstat_party %>%
  select(group,populism) %>% unique() %>% arrange(desc(populism))
kable(dfm_dict1_tstat_party_filtered)
```

group	populism
FDI	0.0266340
M5S	0.0254194
LEGA	0.0231970
MISTO	0.0212365
REG_LEAGUES	0.0209002
FI	0.0206001
PD	0.0201481
LEU	0.0197017
CI	0.0166472
IV	0.0161222
INDIPENDENTE	0.0154037

Rooduijn_Pauwels_Italian

Level of sparsity

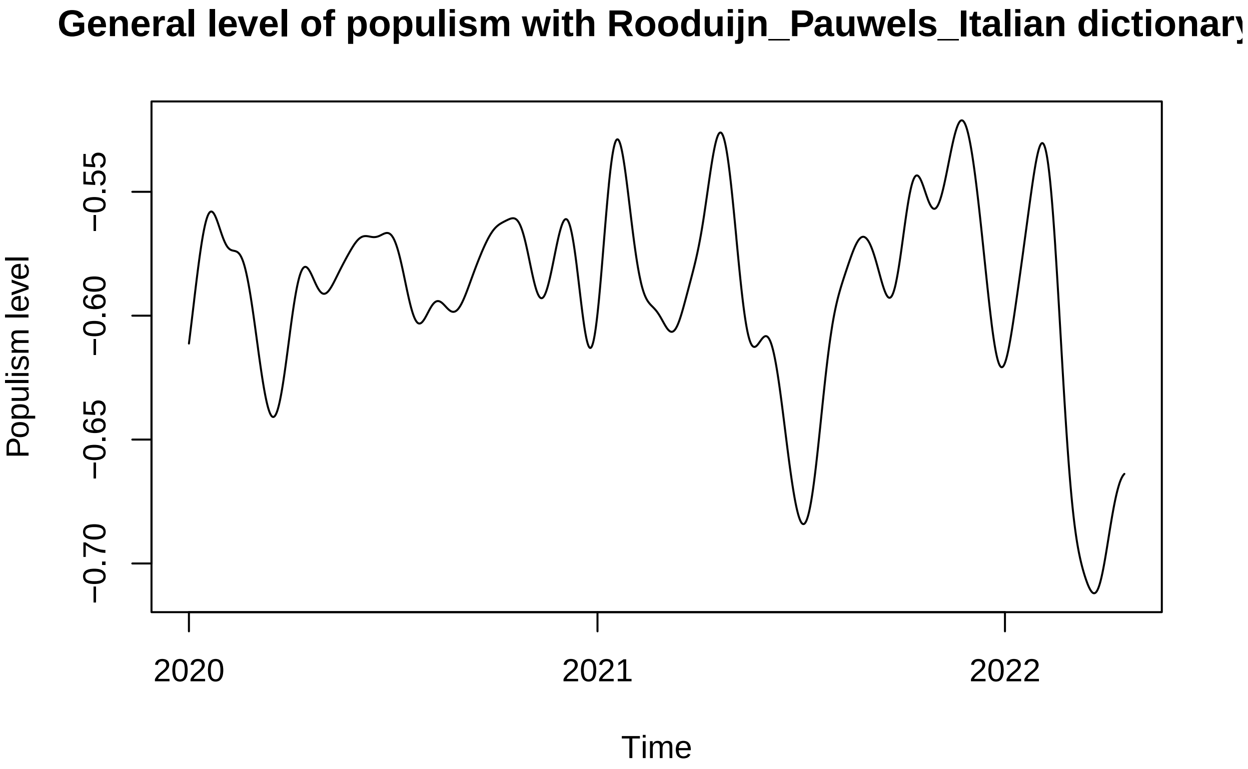
daily: 0.60%

weekly: 0.%

monthly: 0%

```
# Dictionary analysis with Rooduijn_Pauwels_Italian
# By quarter
dfm_dict2  <- dfm_lookup(dfm_weigh_p_quart, dictionary = Rooduijn_Pauwels_Italian)
# Group by date
dfm_by_date2 <- dfm_lookup(dfm_weigh_p_day, dictionary = Rooduijn_Pauwels_Italian)
#dfm_by_date2

# Group by week
#dfm_by_week2 <- dfm_group(dfm_dict2, groups= week)
#dfm_by_week2
# Group by month
#dfm_by_month2 <- dfm_group(dfm_dict2, groups= month)

#kable(dfm_by_month2)
```

General level of populism in time



Most populist parliamentary group

```
# Most populist parliamentary group
dfm_dict2_tstat_party <- textstat_frequency(dfm_dict2, groups = party_id)
kable(dfm_dict2_tstat_party %>% slice_max(frequency, n = 20))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0274375	1	10	FDI
6	populism	0.0215939	1	10	LEGA
7	populism	0.0159693	1	10	LEU
9	populism	0.0156860	1	10	MISTO
10	populism	0.0149336	1	10	PD
8	populism	0.0144550	1	10	M5S
3	populism	0.0116489	1	10	FI
11	populism	0.0083365	1	10	REG_LEAGUES
5	populism	0.0056664	1	9	IV
1	populism	0.0040565	1	9	CI
4	populism	0.0002734	1	1	INDIPENDENTE

Distribution of parliamentary group populism

```
summary(dfm_dict2_tstat_party$frequency)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0002734 0.0070015 0.0144550 0.0127324 0.0158277 0.0274375
```

```
#TBD
hist(dfm_dict2_tstat_party$frequency, prob=T)
points(density(dfm_dict2_tstat_party$frequency),type="l",col="blue")
rug(dfm_dict2_tstat_party$frequency,col="red")

m <- mean(dfm_dict2_tstat_party$frequency)
std<-sqrt(var(dfm_dict2_tstat_party$frequency))

hist(dfm_dict2_tstat_party$frequency,prob=T,main="Frequency")

curve(dnorm(x, mean=m, sd=std), col="darkblue", lwd=2, add=TRUE)
```

```
# above the median
kable(dfm_dict2_tstat_party  %>% filter(frequency > median(frequency)))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0274375	1	10	FDI
6	populism	0.0215939	1	10	LEGA
7	populism	0.0159693	1	10	LEU
9	populism	0.0156860	1	10	MISTO
10	populism	0.0149336	1	10	PD

```
# above the mean
kable(dfm_dict2_tstat_party %>% filter(frequency > mean(frequency)))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0274375	1	10	FDI
6	populism	0.0215939	1	10	LEGA
7	populism	0.0159693	1	10	LEU
8	populism	0.0144550	1	10	M5S
9	populism	0.0156860	1	10	MISTO
10	populism	0.0149336	1	10	PD

```
# below the first quantiles
kable(dfm_dict2_tstat_party %>% filter(frequency < 0.0070015))
```

	feature	frequency	rank	docfreq	group
1	populism	0.0040565	1	9	CI
4	populism	0.0002734	1	1	INDIPENDENTE
5	populism	0.0056664	1	9	IV

```
# above the third quantiles
kable(dfm_dict2_tstat_party %>% filter(frequency > 0.0158277))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0274375	1	10	FDI
6	populism	0.0215939	1	10	LEGA
7	populism	0.0159693	1	10	LEU

Grundl_Italian_adapted

Level of sparsity

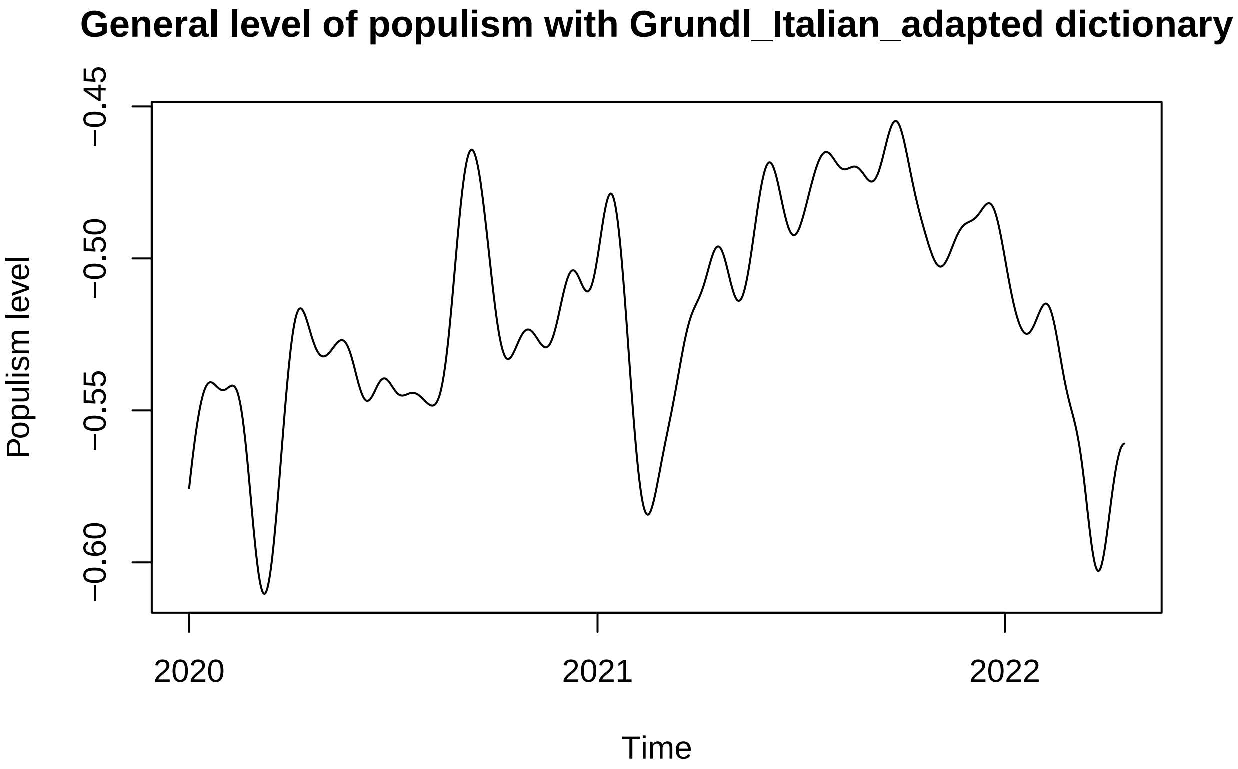
daily: 0.24%

weekly: 0.0%

monthly: 0%

```r
# Dictionary analysis with Grundl_Italian_adapted
# By quarter
dfm_dict3  <- dfm_lookup(dfm_weigh_p_quart, dictionary = Grundl_Italian_adapted)
# Group by date
dfm_by_date3 <- dfm_lookup(dfm_weigh_p_day, dictionary = Grundl_Italian_adapted)
#dfm_by_date2
```

General level of populism in time



Most populist parliamentary group

```r
# Most populist parliamentary group
dict_3_tstat_party <- textstat_frequency(dfm_dict3, groups = party_id)
```

```
kable(dict_3_tstat_party %>% slice_max(frequency, n = 20))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0254825	1	10	FDI
8	populism	0.0231987	1	10	M5S
9	populism	0.0227410	1	10	MISTO
6	populism	0.0224236	1	10	LEGA
3	populism	0.0193174	1	10	FI
7	populism	0.0188197	1	10	LEU
10	populism	0.0174278	1	10	PD
1	populism	0.0160156	1	10	CI
11	populism	0.0108807	1	10	REG_LEAGUES
4	populism	0.0105241	1	8	INDIPENDENTE
5	populism	0.0080644	1	9	IV

Distribution of parliamentary grouppopulism

```
#TBD
summary(dict_3_tstat_party$frequency)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## 0.008064 0.013448 0.018820 0.017718 0.022582 0.025483
```

```
# above the median
kable(dict_3_tstat_party  %>% filter(frequency > median(frequency)))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0254825	1	10	FDI
3	populism	0.0193174	1	10	FI
6	populism	0.0224236	1	10	LEGA
8	populism	0.0231987	1	10	M5S
9	populism	0.0227410	1	10	MISTO

```
# above the mean
kable(dict_3_tstat_party %>% filter(frequency > mean(frequency)))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0254825	1	10	FDI
3	populism	0.0193174	1	10	FI
6	populism	0.0224236	1	10	LEGA
7	populism	0.0188197	1	10	LEU
8	populism	0.0231987	1	10	M5S
9	populism	0.0227410	1	10	MISTO

```
# below the first quantiles
kable(dict_3_tstat_party %>% filter(frequency < 0.013448))
```

	feature	frequency	rank	docfreq	group
4	populism	0.0105241	1	8	INDIPENDENTE
5	populism	0.0080644	1	9	IV
11	populism	0.0108807	1	10	REG_LEAGUES

```
# above the third quantiles
kable(dict_3_tstat_party %>% filter(frequency > 0.022582 ))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0254825	1	10	FDI
8	populism	0.0231987	1	10	M5S
9	populism	0.0227410	1	10	MISTO

Decadri_Boussalis

Level of sparsity

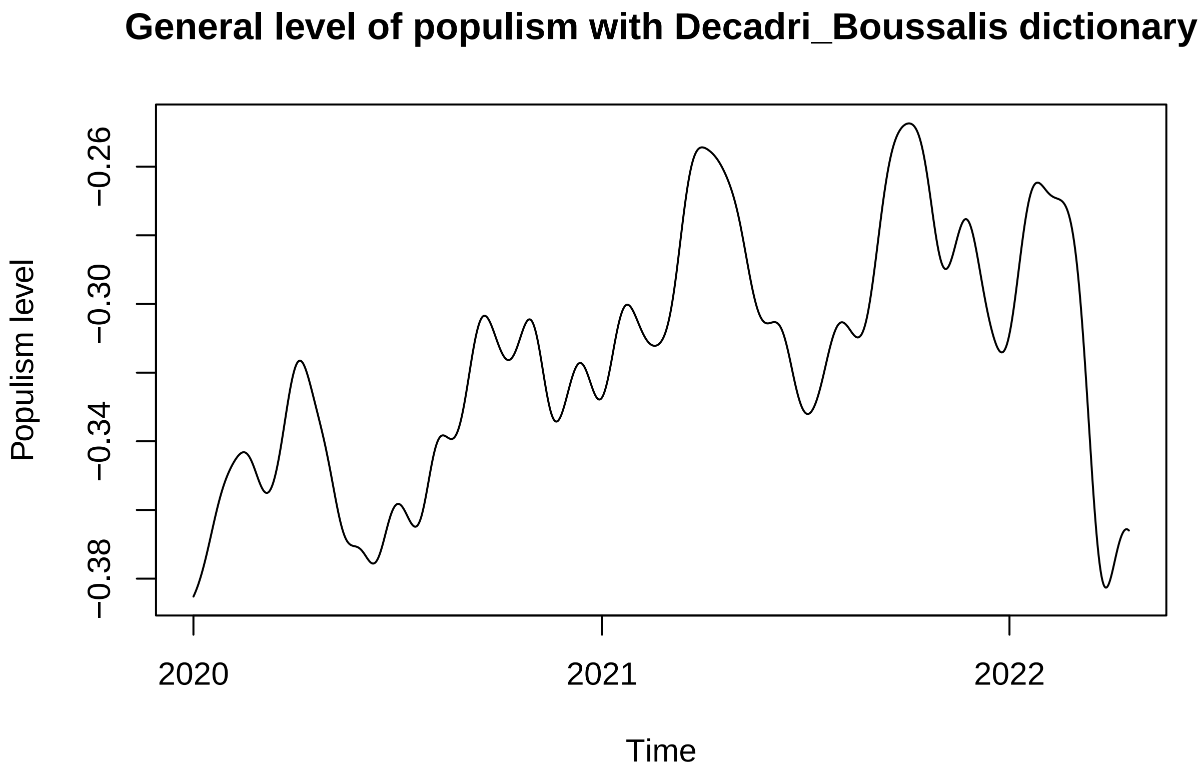
daily: 0%

weekly: 0.0%

monthly: 0%

```
# Dictionary analysis with Decadri_Boussalis
# By quarter
dfm_dict4  <- dfm_lookup(dfm_weigh_p_quart, dictionary = Decadri_Boussalis)
# By date
dfm_by_date4 <- dfm_lookup(dfm_weigh_p_day, dictionary = Decadri_Boussalis)
```

General level of populism in time



Most populist parliamentary group

```
# Most populist parliamentary group
dict_4_tstat_party <- textstat_frequency(dfm_dict4, groups = party_id)

kable(dict_4_tstat_party %>% slice_max(frequency, n = 20))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0781517	1	10	FDI
8	populism	0.0696170	1	10	M5S
6	populism	0.0656655	1	10	LEGA
3	populism	0.0600724	1	10	FI
9	populism	0.0595844	1	10	MISTO
7	populism	0.0588181	1	10	LEU
10	populism	0.0586910	1	10	PD
1	populism	0.0499145	1	10	CI
5	populism	0.0479199	1	10	IV
4	populism	0.0464945	1	10	INDIPENDENTE
11	populism	0.0435615	1	10	REG_LEAGUES

Distribution of party populism

```
#TBD
summary(dict_4_tstat_party$frequency)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04356 0.04892 0.05882 0.05804 0.06287 0.07815
```

```
# above the median
kable(dict_4_tstat_party  %>% filter(frequency > median(frequency)))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0781517	1	10	FDI
3	populism	0.0600724	1	10	FI
6	populism	0.0656655	1	10	LEGA
8	populism	0.0696170	1	10	M5S
9	populism	0.0595844	1	10	MISTO

```
# above the mean
kable(dict_4_tstat_party %>% filter(frequency > mean(frequency)))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0781517	1	10	FDI
3	populism	0.0600724	1	10	FI
6	populism	0.0656655	1	10	LEGA
7	populism	0.0588181	1	10	LEU
8	populism	0.0696170	1	10	M5S
9	populism	0.0595844	1	10	MISTO
10	populism	0.0586910	1	10	PD

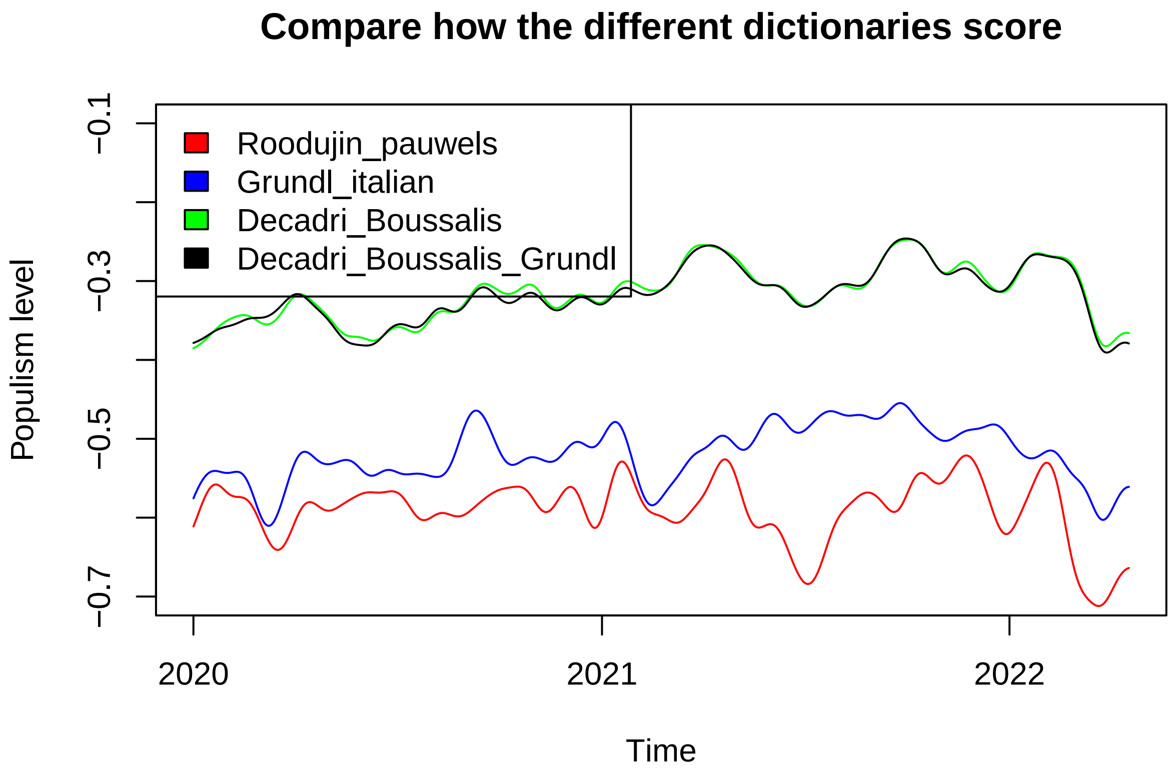
```
# below the first quantiles
kable(dict_4_tstat_party %>% filter(frequency < 0.04892 ))
```

	feature	frequency	rank	docfreq	group
4	populism	0.0464945	1	10	INDIPENDENTE
5	populism	0.0479199	1	10	IV
11	populism	0.0435615	1	10	REG_LEAGUES

```
# above the third quantiles
kable(dict_4_tstat_party %>% filter(frequency > 0.06287  ))
```

	feature	frequency	rank	docfreq	group
2	populism	0.0781517	1	10	FDI
6	populism	0.0656655	1	10	LEGA
8	populism	0.0696170	1	10	M5S

Compare the general level of populism in time for the dictionaries



Compare how the dictionaries score for the most populist parliamentary group

```
# Create the columns with the "populist score"
# 11 for the "most populist" and 1 for the least
dfm_dict1_tstat_party_filtered$my_rank <- rank(dfm_dict1_tstat_party_filtered$populism)
dfm_dict2_tstat_party$my_rank <- rank(dfm_dict2_tstat_party$frequency)
dict_3_tstat_party$my_rank <- rank(dict_3_tstat_party$frequency)
dict_4_tstat_party$my_rank <- rank(dict_4_tstat_party$frequency)

# define the parliamentary group list
party <- c("LEGA","PD","M5S","FI","FDI","MISTO",
           "LEU","CI","IV","INDIPENDENTE","REG_LEAGUES")
```

```
# create an empty df
party_rank <- data.frame(first = vector(), second = vector(),
                         third = vector(), fourth = vector(), fifth = vector() )

# loop the rank for each parliamentary group
for (i in party)
{
  rank_dict_1 <- (dfm_dict1_tstat_party_filtered %>% filter(group == i ) %>% .$my_rank)
  rank_dict_2 <- (dfm_dict2_tstat_party %>% filter(group == i ) %>% .$my_rank)
  rank_dict_3 <- (dict_3_tstat_party %>% filter(group == i ) %>% .$my_rank)
  rank_dict_4 <- (dict_4_tstat_party %>% filter(group == i ) %>% .$my_rank)

  party <- (i)
  party_rank <- rbind(party_rank, cbind(party, rank_dict_1, rank_dict_2,
                                        rank_dict_3, rank_dict_4))
}

# change the format of the columns in numeric
party_rank$rank_dict_1 <- as.numeric(party_rank$rank_dict_1)
party_rank$rank_dict_2 <- as.numeric(party_rank$rank_dict_2)
party_rank$rank_dict_3 <- as.numeric(party_rank$rank_dict_3)
party_rank$rank_dict_4 <- as.numeric(party_rank$rank_dict_4)

# Create the column with the sum of the single score
party_rank$total_score <- rowSums(party_rank[,-1])
kable(party_rank %>% arrange(desc(total_score)))
```

party	rank_dict_1	rank_dict_2	rank_dict_3	rank_dict_4	total_score
FDI	11	11	11	11	44
LEGA	9	10	8	9	36
M5S	10	6	10	10	36
MISTO	8	8	9	7	32
FI	6	5	7	8	26
LEU	4	9	6	6	25
PD	5	7	5	5	22
REG_LEAGUES	7	4	3	1	15
CI	3	2	4	4	13
IV	2	3	1	3	9
INDIPENDENTE	1	1	2	2	6