# CDU/CSU in Their New Role as Opposition - Using Populist Communication as a Mainstream Party

**Jana Jung** *Univeristy of Mannheim*

In this analysis the CDU/CSU Parliamentary Group was used as a case to analyze how the degree of populist communication differs between being in government or in opposition, using social media data from Twitter and a dictionary-based measurement of populist communication. Additionally, it was studied how the potential change in populist communication influenced the popularity of posts on social media. It was found that populist tweets were more likely when in opposition then when in government, resulting in a more populist communication after the federal election in 2021. As expected, there were no differences in the degree of populist communication between the CDU and the CSU. When analyzing the popularity cues of tweets, it was found that populist tweets received a higher amount of popularity cues than non-populist tweets. Based on this result, the party's strategical change in populist communication on Twitter can be considered successful. However, data collection was constricted due to restrictions of the Twitter API which might limit the informative value of this study.

### Hypotheses

> **H1**: When in opposition, the communication of the CDU/CSU Parliamentary Group is more populist than when in government.
>
> **H2**: There is no difference in the degree of populist communication between the CDU and the CSU.
>
> **H3**: Whether a tweet contains populist content or not has an influence on its popularity cues.

### Data and Methods

In order to test the presented hypotheses, social media data from Twitter and a dictionary-based measurement of populist communication were used. Tweets from relevant politicians of the CDU and the CSU were collected from a two year period frame surrounding the German federal election on the 26 September 2021. To analyze the collected data, the dictionary by @grundl_populist_2022 was applied.

```
library(rtweet)
library(tidyverse)
library(quanteda)
library(popdictR)
library(oolong)
library(ggplot2)
library(gridExtra)
library(knitr)
```

```r
library(stats)
library(rstatix)
```

*Data*

Social media data from Twitter was collected through the Twitter API.

```r
auth_as("default")

acc_cdu <- c("_FriedrichMerz",
             "akk",
             "MarioCzaja",
             "ChristinaStumpp",
             "maxmoerseburg",
             "PaulZiemiak",
             "ArminLaschet")

acc_csu <- c("Markus_Soeder",
             "MartinHuberCSU",
             "csu_bt",
             "DoroBaer",
             "Brehm_inNBGNord",
             "HPFriedrichCSU",
             "DaniLudwigMdB",
             "MichaelFrieser")


tweets_cdu <- get_timeline(user = acc_cdu,
                           n = Inf,
                           retryonratelimit = TRUE,
                           include_rts = TRUE)

tweets_csu <- get_timeline(user = acc_csu,
                           n = Inf,
                           retryonratelimit = TRUE,
                           include_rts = TRUE)


# save tweets
save(tweets_cdu, file = "tweets_cdu")
save(tweets_csu, file = "tweets_csu")
```

```r
# load tweets
load("tweets_cdu")
load("tweets_csu")
```

*Preprocessing*

```r
# prepocessing the data
tweets_cdu_clean <- tweets_cdu %>%
  select(created_at,
         id,
         full_text,
         retweet_count,
         favorite_count,
         lang) %>%
  rename(doc_id = id,
         text = full_text) %>%
  # only keep tweets from 26.09.2020 to 26.09.2022
  filter(created_at >= as.POSIXct("2020-09-26 00:00:00", tz="Europe/Berlin"),
         created_at < as.POSIXct("2022-09-27 00:00:00", tz="Europe/Berlin")) %>%
  # only keep tweets in German
  filter(lang == "de") %>%
  # only keep tweets which are not retweets
  filter(grepl("^RT", text) == FALSE) %>%
  # create new variables "party", "role", and "popularity_cues"
  mutate(party = "CDU",
         role = case_when(created_at < as.POSIXct("2021-09-27 00:00:00",
                                                  tz="Europe/Berlin") ~ "government",
                          created_at >= as.POSIXct("2021-09-27 00:00:00",
                                                  tz="Europe/Berlin") ~ "opposition"),
         popularity_cues = retweet_count + favorite_count)


tweets_csu_clean <- tweets_csu %>%
  select(created_at,
         id,
         full_text,
         retweet_count,
         favorite_count,
         lang) %>%
  rename(doc_id = id,
         text = full_text) %>%
  # filter tweets to only keep the ones from 26.09.2020 to 26.09.2022
  filter(created_at >= as.POSIXct("2020-09-26 00:00:00", tz="Europe/Berlin"),
         created_at < as.POSIXct("2022-09-27 00:00:00", tz="Europe/Berlin")) %>%
  # only keep tweets in German
  filter(lang == "de") %>%
  # only keep tweets which are not retweets
  filter(grepl("^RT", text) == FALSE) %>%
  #create new variables "party", "role", and "popularity_cues"
  mutate(party = "CSU",
         role = case_when(created_at < as.POSIXct("2021-09-27 00:00:00",
```

```
                                               tz="Europe/Berlin") ~ "government",
                          created_at >= as.POSIXct("2021-09-27 00:00:00",
                                               tz="Europe/Berlin") ~ "opposition"),
         popularity_cues = retweet_count + favorite_count)

tweets_all <- rbind(tweets_cdu_clean, tweets_csu_clean)

tweets_corp <- corpus(tweets_all)
```

*Measuring Populism Using Dictionary Analysis*

```
# dictionary application
tweets_dict <- tweets_corp %>%
  run_popdict(at_level = "documents", return_value = "binary") %>%
  convert(to = "data.frame") %>%
  rename(populist = dict_gruendl_2020)
```

*Dictionary Validation*

To validate the results of the dictionary analysis, a random sample of 80 tweets were coded manually.

```
# create new data frame for validation
tweets_val <- rbind(tweets_cdu, tweets_csu) %>%
  # filter tweets to only keep the ones outside of the timeframe
  filter(created_at < as.POSIXct("2020-09-26 00:00:00", tz="Europe/Berlin") |
         created_at >= as.POSIXct("2022-09-27 00:00:00", tz="Europe/Berlin")) %>%
  # only keep tweets in German
  filter(lang == "de") %>%
  # only keep tweets which are not retweets
  filter(grepl("^RT", text) == FALSE) %>%
  select(created_at,
         id,
         full_text) %>%
  rename(doc_id = id,
         text = full_text)
```

Manual coding was performed once and the results were saved for later use.

```
# perform manual coding
gs_test <- create_oolong(input_corpus = tweets_val$text,
                         construct = "populist",
                         exact_n = 80)

gs_test$do_gold_standard_test()
gs_test$lock()
```

```
save(gs_test, file = "gs_test")
```

Afterwards, the dictionary was applied to the same tweets from the random sample. The comparison is shown in Figure 1. It can be seen that only one tweet was rated differently. However, it should be noted that populist tweets were very rare. Therefore, the data set was highly unbalanced.

```
load("gs_test")

# get data frame containing the answers of the human coder
gs_corpus <- gs_test$turn_gold()
gs_coded <-
  gs_corpus %>%
  convert(to = "data.frame") %>%
  select(text,
         answer)

# change answer to either 0 or 1 to match with the variable populist
gs_coded$answer[gs_coded$answer == 1] <- 0
gs_coded$answer[gs_coded$answer == 5] <- 1

# get all tweets of the sample
gs_data <-
  tweets_val %>%
  filter(text %in% gs_coded$text) %>%
  corpus()

# apply the dictionary on the tweets from the random sample
gs_scored <- gs_data %>%
  run_popdict(at_level = "documents", return_value = "binary") %>%
  convert(to = "data.frame") %>%
  rename(populist = dict_gruendl_2020)

gs_comparison <- merge(gs_coded, gs_scored, by.x = "text")

ggplot(gs_comparison, aes(x=as.factor(populist), y=as.factor(answer))) +
  geom_jitter(width=0.1, height=0.1, color = "#669933") +
  scale_x_discrete(name = "Manual Rating", breaks = c(0, 1)) +
  scale_y_discrete(name = "Dictionary's Result", breaks = c(0, 1)) +
  theme_minimal()
```

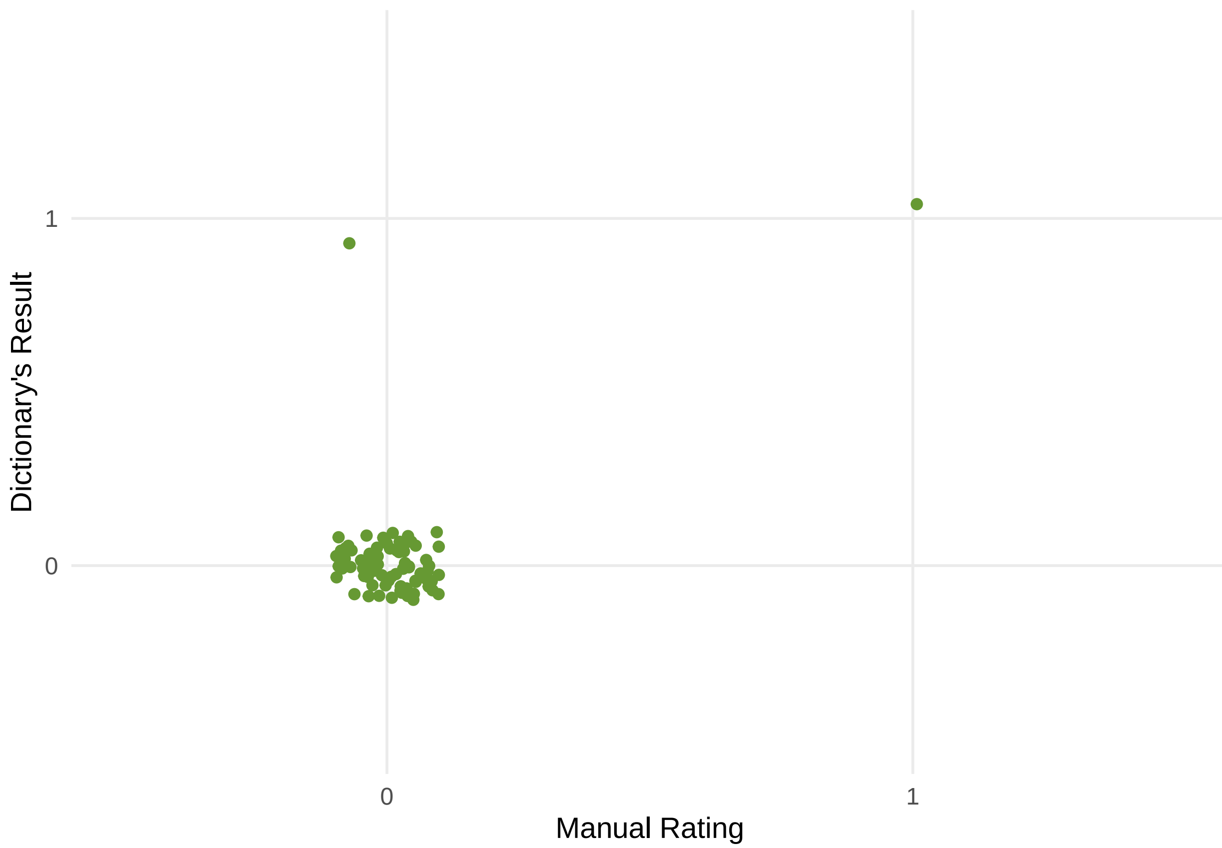


Figure 1: Comparison of the Manual Ratings and the Results of the dictionary - Populist tweets are rated as 1 and non-populist tweets are rated as 0

**Results**

*Descriptives*

```
kable(table(tweets_dict$populist),
      col.names = c("Populist", "Frequency"),
      caption = "Frequency of Tweets Considered as Non-Populist (0) and
      as Populist (1)")
```

Table 1: Frequency of Tweets Considered as Non-Populist (0) and as Populist (1)

| Populist | Frequency |
|---|---|
| 0 | 7689 |
| 1 | 197 |

```
# check percentage frequency distributions
party_freq <- ggplot(data = tweets_dict) +
  geom_bar(mapping = aes(x = party, y = ..prop.., group = 1), stat = "count",
           fill="#669933") +
  scale_y_continuous(name = "Percent", labels = scales::percent_format(),
                     limits = c(0, 1)) +
  xlab("Party") +
  theme_minimal()

role_freq <- ggplot(data = tweets_dict) +
  geom_bar(mapping = aes(x = role, y = ..prop.., group = 1), stat = "count",
           fill="#FFCC33") +
  scale_y_continuous(name = NULL, labels = scales::percent_format(),
                     limits = c(0, 1)) +
  xlab("Role") +
  theme_minimal()

pop_freq <- ggplot(data = tweets_dict) +
  geom_bar(mapping = aes(x = factor(populist), y = ..prop.., group = 1),
           stat = "count", fill="#CC9900") +
  scale_y_continuous(name = NULL, labels = scales::percent_format(),
                     limits = c(0, 1)) +
  scale_x_discrete(name = "Populist", breaks = c(0, 1),
                   labels = c("no", "yes")) +
  theme_minimal()

grid.arrange(party_freq, role_freq, pop_freq, ncol = 3)
```

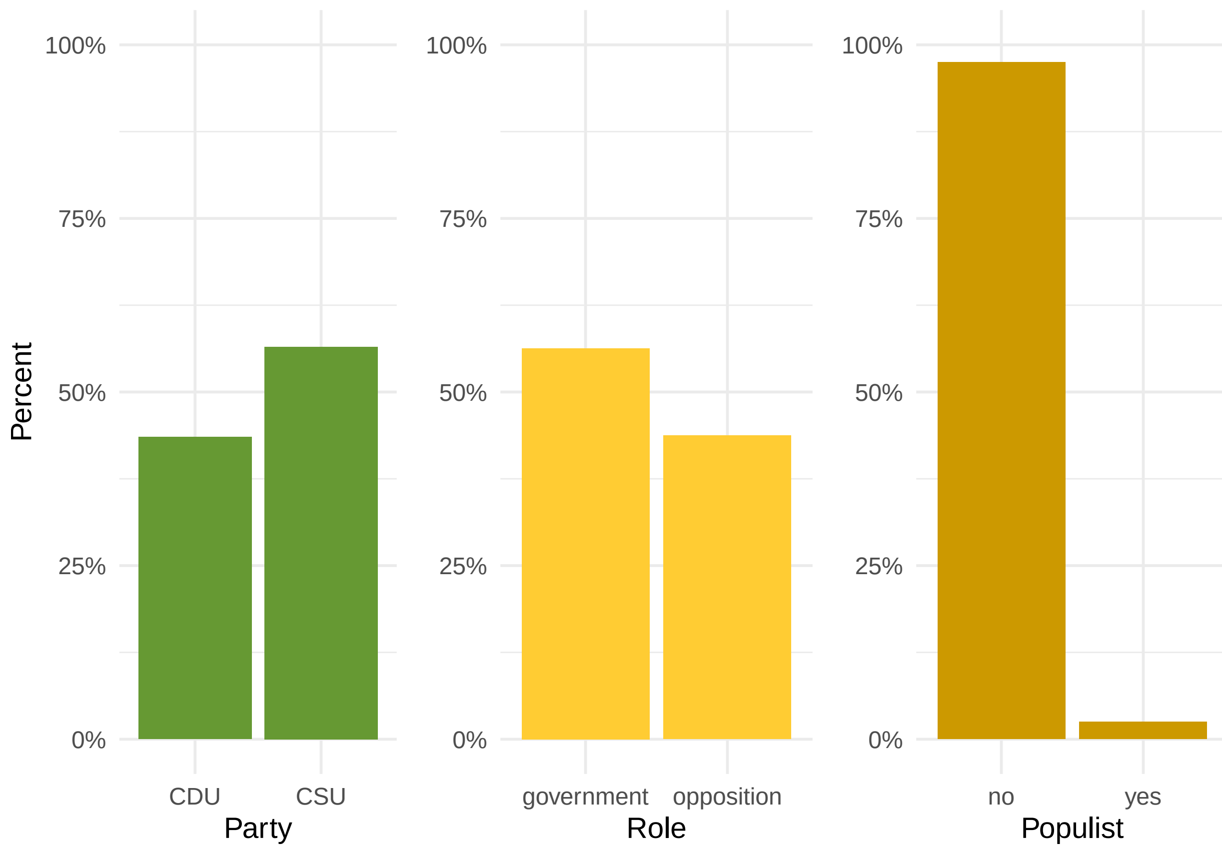


Figure 2: Percentage Frequency Distribution of the Three Dichotomous Variables Party, Role, and Populist

```r
# check distributions of popularity_cues
mean(tweets_dict$popularity_cues)
```

```
## [1] 391.2007
```

```r
sd(tweets_dict$popularity_cues)
```

```
## [1] 1030.936
```

```r
min(tweets_dict$popularity_cues)
```

```
## [1] 0
```

```r
max(tweets_dict$popularity_cues)
```

```
## [1] 29222
```

```r
ggplot(tweets_dict, aes(x=popularity_cues)) +
  geom_histogram(fill="#669933") +
  xlab("Popularity Cues") +
  ylab("Count") +
  theme_minimal()
```

*Hypothesis 1*

In order to test Hypothesis 1, it was analyzed if there was a significant relationship between the variables `role` and `populist`, expecting more populist tweets when in the role of the opposition. Looking at the frequency of populist tweets by role, it can be seen that the percentage of populist tweets is higher for opposition than for government (see Table 2). For statistical hypothesis testing a McNemar's Chi-squared test was used because the two samples (government vs. opposition) are not independent. The difference between government and opposition is simply a change over time while still collecting tweets from the same politicians. The relationship between the variables `role` and `populist` was significant, $\chi^2(1, N = 7886) = 3111.2$, $p < .001$. Populist tweets were more likely when in opposition then when in government.

```r
summaryH1 <- tweets_dict %>%
  group_by(role) %>%
  summarise(
    tweets = n(),
    absolut = sum(populist),
    percent = absolut / tweets * 100) %>%
  ungroup()

kable(summaryH1,
      col.names = c("Role", "Total # of Tweets", "n", "%"),
      caption = "Frequency of Populist Tweets by Role",
      digits = 2)
```

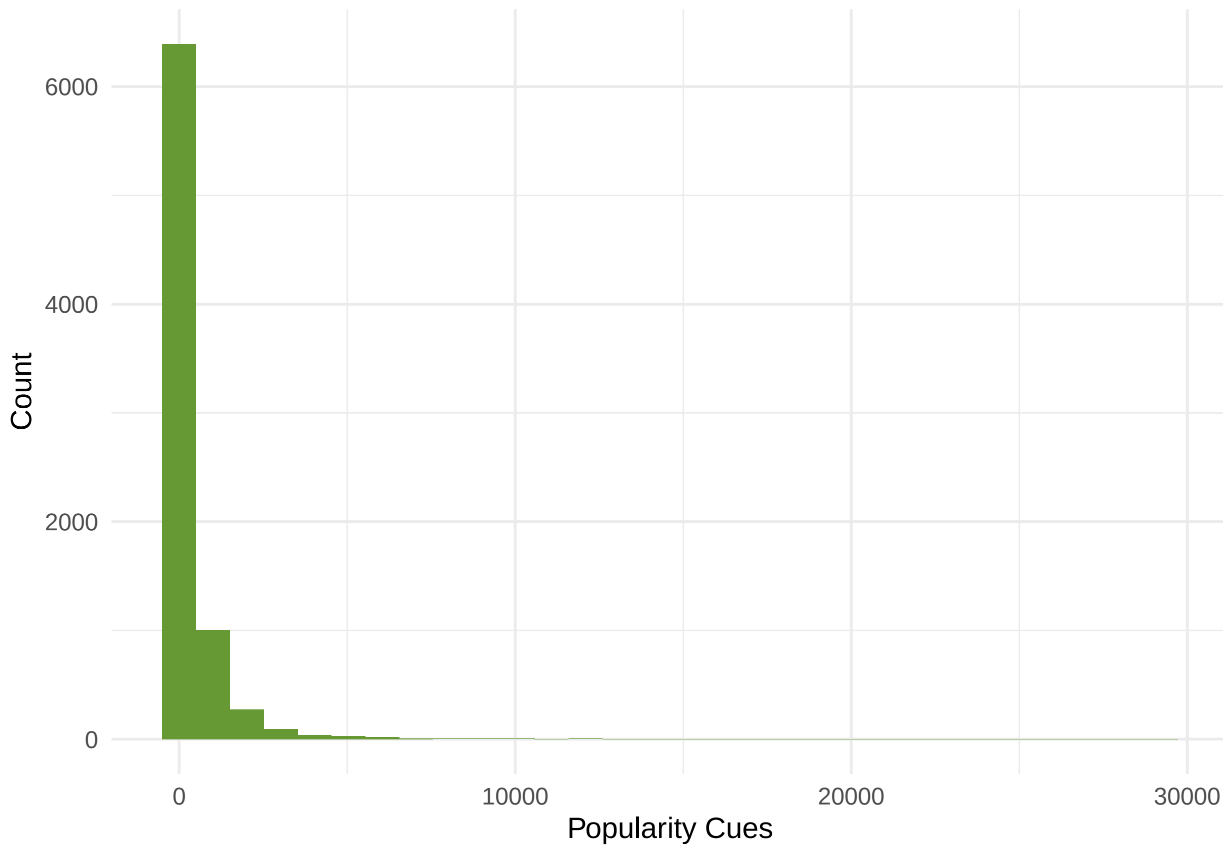


Figure 3: Histogram of the Variable Popularity Cues

Table 2: Frequency of Populist Tweets by Role

| Role | Total # of Tweets | n | % |
|---|---|---|---|
| government | 4438 | 72 | 1.62 |
| opposition | 3448 | 125 | 3.63 |

```
# Hypothesis 1 - McNemar's Chi-squared test
H1 <- mcnemar.test(tweets_dict$role, tweets_dict$populist)
```

*Hypothesis 2*

In order to test Hypothesis 2, it was analyzed if there was a significant relationship between the variables `party` and `populist`, expecting no significant association. Looking at the frequency of populist tweets by party, it can be seen that the percentage of populist tweets is slightly higher for the CSU than for the CDU (see Table 3). For statistical hypothesis testing a Chi-squared test was used. The relationship between the variables `party` and `populist` was not significant, $\chi^2(1, N = 7886) = 2.21$, $p = .14$.

```
summaryH2 <- tweets_dict %>%
  group_by(party) %>%
  summarise(
    tweets = n(),
    absolut = sum(populist),
    percent = absolut / tweets * 100) %>%
  ungroup()

kable(summaryH2,
      col.names = c("Party", "Total # of Tweets", "n", "%"),
      caption = "Frequency of Populist Tweets by Party",
      digits = 2)
```

Table 3: Frequency of Populist Tweets by Party

| Party | Total # of Tweets | n | % |
|---|---|---|---|
| CDU | 3431 | 75 | 2.19 |
| CSU | 4455 | 122 | 2.74 |

```
# Hypothesis 2 - Chi-Squared test
H2 <- chisq.test(tweets_dict$party, tweets_dict$populist)
```

*Hypothesis 3*

In order to test Hypothesis 3, it was analyzed whether or non populist and non-populist tweets received a different amount of popularity cues. For statistical hypothesis testing a two-tailed t-test was used. There was a significant effect of the variable `populist` on the amount of popularity cues received, $t(197) = -2.5$, $p = 0.01$. Populist tweets received a higher amount of popularity cues than non-populist tweets (see Table 4).

```r
summaryH3 <- tweets_dict %>%
  group_by(populist) %>%
  summarise(
    tweets = n(),
    popularity_cues_mean = mean(popularity_cues),
    popularity_cues_sd = sd(popularity_cues)) %>%
  ungroup()

kable(summaryH3,
      col.names = c("Populist", "Total # of Tweets", "M", "SD"),
      caption = "Mean and SD of the Amount of Popularity Cues for Non-Populist (0)
      and Populist Tweets (1) ",
      digits = 2)
```

Table 4: Mean and SD of the Amount of Popularity Cues for Non-Populist (0) and Populist Tweets (1)

| Populist | Total # of Tweets | M | SD |
|---:|---:|---:|---:|
| 0 | 7689 | 379.29 | 950.04 |
| 1 | 197 | 856.09 | 2670.50 |

```r
# Hypothesis 3 - t-test
H3 <- tweets_dict %>%
  t_test(popularity_cues ~ populist)
```

**Summary**

In this analysis, the CDU/CSU Parliamentary Group was used as a case to analyse how the degree of populist communication differs between being in government or in opposition, using social media data from Twitter and a dictionary-based measurement of populist communication. Additionally, it was studied how the potential change in populist communication influences the popularity of posts on social media.

In summary, populist tweets were more likely when in opposition then when in government, resulting in a more populist communication after the federal election in 2021. As expected, there were no differences in the degree of populist communication between the CDU and the CSU. When analyzing the popularity cues of tweets, it was found that populist tweets received a higher amount of popularity cues than non-populist tweets. Based on this result, the party's strategical change in populist communication on Twitter can be considered successful.