Preparing the textual data

Cornelius Erfort

6/3/2021

Contents

1 Setting up

This script requires the file "sample_germany.dta" which is not included on GitHub.

1.1 Loading packages

This script is based mainly on the functions of the quanteda package. For the cross-validation of the textmodels, quanteda.classifiers has to be loaded from GitHub.

```
start_time <- Sys.time()

packages <- c("quanteda", "dplyr", "tm", "rmarkdown", "plyr", "readr", "ggplot2",
    "stringr", "formatR", "readstata13", "lubridate", "kableExtra", "stargazer")

lapply(packages[!(packages %in% rownames(installed.packages()))], install.packages)
invisible(lapply(packages, require, character.only = T))

theme_update(text = element_text(family = "LM Roman 10"))  # Set font family for ggplot
```

1.2 Loading data

The sample data for Germany consists of 2,740 labeled press releases. The dataset is not uploaded on GitHub.

```
sample_germany <- read.dta13("data/sample_germany.dta", convert.factors = F)

# Correcting classification for three documents (cat 21)
sample_germany$issue[sample_germany$id == 229] <- 191
sample_germany$issue[sample_germany$id == 731] <- 7
sample_germany$issue[sample_germany$id == 902] <- 10

sample_germany <- filter(sample_germany, date != "NA" | !is.na(text))
nrow(sample_germany)
```

```
## [1] 2740
```

issue	1	2	3	4	5	6	7	8	9	10	12	13	14	15	16	17	18	20	23	99	191	192
n	175	181	119	99	167	137	84	105	131	74	195	104	32	168	121	68	27	97	19	44	350	152

```
# Subset to relevant vars
germany_textpress <- sample_germany %>%
    select("header", "text", "issue", "position", "id", "party", "date")

# Remove non-thematic press releases
germany_textpress <- germany_textpress %>%
    filter(issue != 98)

# Distribution of issues in the hand-coded sample
table(germany_textpress$issue) %>%
    as.data.frame() %>%
    dplyr::rename(issue = Var1, n = Freq) %>%
    t() %>%
    kbl(booktabs = T) %>%
    kable_styling(latex_options = "scale_down")
```

1.3 Merging categories

In order to improve the classification, similar topics are merged or subsumed under the "Other" category. In practice, press releases regarding, for instance, Environment and Energy are often not distinguishable. Furthermore, small categories with very few observations are not suitable for automated classification.

```
germany_textpress$issue_r1 <- as.numeric(germany_textpress$issue)

# Merge categories
germany_textpress <- germany_textpress %>% mutate(issue_r1 = recode(issue_r1,
                           `8`  = 7,  # Environment & Energy
                           `13` = 10, # Transportation & Welfare
                           `14` = 10, # Housing & Welfare
                           `18` = 15, # Foreign Trade and Domestic Commerce
                           `23` = 99) # Culture: Too few observations
                                                )
# Category descriptions
issue_categories <-
  data.frame(issue_r1 = c(1:7, 9:10, 12, 15:17, 20, 99, 191:192),
             issue_r1_descr = c("Macroeconomics", "Civil Rights",
                                "Health", "Agriculture", "Labor", "Education", "Environment and Energy"
                                "Immigration", "Welfare", "Law and Crime", "Commerce", "Defense",
                                "Technology", "Government Operations", "Other", "International Affairs"
save(issue_categories, file = "supervised-files/issue_categories.RData")

issue_categories %>% dplyr::rename("Issue number" = issue_r1, "Issue name" = issue_r1_descr) %>%
  kbl(booktabs = T)
```

Issue number	Issue name
1	Macroeconomics
2	Civil Rights
3	Health
4	Agriculture
5	Labor
6	Education
7	Environment and Energy
9	Immigration
10	Welfare
12	Law and Crime
15	Commerce
16	Defense
17	Technology
20	Government Operations
99	Other
191	International Affairs
192	EU

```
# Write latex table
if(!dir.exists("tables")) dir.create("tables")
issue_categories_out <- issue_categories[c(1:13, 16:17, 14:15), ]
issue_categories_out$issue_r1 <- as.character(issue_categories_out$issue_r1)
issue_categories_out$issue_r1[issue_categories_out$issue_r1 == "191"] <- "19.1"
issue_categories_out$issue_r1[issue_categories_out$issue_r1 == "192"] <- "19.2"
issue_categories_out %>%
  dplyr::rename(Code = issue_r1, Topic = issue_r1_descr) %>%
  stargazer(out = "tables/issue_categories.tex", summary = F, rownames = F,
            title = "Issue categories used for classification",
            label = "tab:issue_categories")
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Thu, Jun 03, 2021 - 12:16:41
## \begin{table}[!htbp] \centering
##   \caption{Issue categories used for classification}
##   \label{tab:issue_categories}
## \begin{tabular}{@{\extracolsep{5pt}} cc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Code & Topic \\
## \hline \\[-1.8ex]
## 1 & Macroeconomics \\
## 2 & Civil Rights \\
## 3 & Health \\
## 4 & Agriculture \\
## 5 & Labor \\
## 6 & Education \\
## 7 & Environment and Energy \\
## 9 & Immigration \\
## 10 & Welfare \\
## 12 & Law and Crime \\
```

issue	1	2	3	4	5	6	7	9	10	12	15	16	17	20	99	191	192
n	175	181	119	99	167	137	189	131	210	195	195	121	68	97	63	350	152

```
## 15 & Commerce \\
## 16 & Defense \\
## 17 & Technology \\
## 19.1 & International Affairs \\
## 19.2 & EU \\
## 20 & Government Operations \\
## 99 & Other \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```
# Distribution with merged categories
table(germany_textpress$issue_r1) %>% as.data.frame() %>%
  dplyr::rename(issue = Var1, n = Freq) %>% t() %>% kbl(booktabs = T) %>%
  kable_styling(latex_options="scale_down")
```

```
# Party names
party_names <- data.frame(party = c(1:8),
                          party_name = c("Bündnis 90/Die Grünen - Fraktion",
                                         "AfD - Bundesverband", "AfD - Fraktion",
                                         "FDP - Bundesverband", "FDP - Fraktion", "DIE LINKE - Fraktion
                                         "SPD - Fraktion", "CDU/CSU - Fraktion"))
germany_textpress <- merge(germany_textpress, party_names, by = "party")

# Distribution by parties
table(germany_textpress$party_name) %>% as.data.frame() %>%
  dplyr::rename(party = Var1, n = Freq) %>% kbl(booktabs = T)
```

party	n
AfD - Bundesverband	106
AfD - Fraktion	11
Bündnis 90/Die Grünen - Fraktion	472
CDU/CSU - Fraktion	368
DIE LINKE - Fraktion	634
FDP - Bundesverband	212
FDP - Fraktion	288
SPD - Fraktion	558

```
table(germany_textpress$party_name, substr(germany_textpress$date, 6, 10)) %>%
  as.data.frame.matrix() %>% kbl(booktabs = T)
```

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
AfD - Bundesverband	0	0	0	4	21	10	20	22	17	12
AfD - Fraktion	0	0	0	0	0	0	0	11	0	0
Bündnis 90/Die Grünen - Fraktion	87	76	59	49	44	43	42	22	38	12
CDU/CSU - Fraktion	58	61	45	50	48	35	38	23	10	0
DIE LINKE - Fraktion	88	78	72	82	64	67	52	57	54	20
FDP - Bundesverband	10	9	10	5	43	32	40	38	16	9
FDP - Fraktion	59	65	62	43	0	0	0	6	33	20
SPD - Fraktion	110	86	93	61	50	50	35	32	29	12

```
germany_textpress$htext <- str_c(germany_textpress$header, " ", germany_textpress$text)

# Make order of documents random
set.seed(4325)
germany_textpress <- germany_textpress[sample(1:nrow(germany_textpress), nrow(germany_textpress)), ]
germany_textpress$cv_sample <- sample(1:5, nrow(germany_textpress), replace = T)

save(germany_textpress, file = "supervised-files/germany_textpress.RData")
```

1.4 Creating the document frequency matrix (dfm)

We create a text corpus based on the header and text of each press release. We draw a random sample from the corpus to create a training and a test dataset. The test dataset consists of approx. one fifth of the documents.

Subsequently, we follow standard procedures for the preparation of the document frequency matrix. First, we remove stopwords and stem the words in order to better capture the similarities across documents. Second, we remove all punctuation, numbers, symbols and URLs. In a last step, we remove all words occurring in less than 0.5% or more than 90% of documents.

```
if(!dir.exists("supervised-files/train-test")) dir.create("supervised-files/train-test")
if(file.exists("supervised-files/train-test/dfmat.RData") & file.exists("supervised-files/train-test/df

  load("supervised-files/train-test/dfmat.RData")
  load("supervised-files/train-test/dfmat_training.RData")
  load("supervised-files/train-test/dfmat_test.RData")

  } else {

  corp_press <- corpus(str_c(germany_textpress$header, " ", germany_textpress$text),
                       docvars = select(germany_textpress, c(id, issue_r1, party_name, cv_sample)))

# Create dfm
dfmat <- corpus_subset(corp_press) %>%
  dfm(remove = stopwords("de"), # Stem and remove stopwords, punctuation etc.
      stem = T,
      remove_punct = T,
      remove_number = T,
      remove_symbols = T,
      remove_url = T) %>%
  dfm_trim(min_docfreq = 0.005, # Remove words occurring <.5% or > 80% of docs
           max_docfreq = .9,
           docfreq_ = "prop") %>%
```

```
  suppressWarnings()

save(dfmat, file = "supervised-files/train-test/dfmat.RData")

# Create training and test set (also as csv for Python)
dfmat_training <- dfm_subset(dfmat, dfmat$cv_sample != 1)
save(dfmat_training, file = "supervised-files/train-test/dfmat_training.RData")
as.data.frame(as.matrix(dfmat_training, verbose = T)) %>%
  write.csv(., "supervised-files/train-test/train.csv")
write.csv(dfmat_training$issue_r1, "supervised-files/train-test/train_val.csv")

dfmat_test <- dfm_subset(dfmat, dfmat$cv_sample == 1)
as.data.frame(as.matrix(dfmat_test, verbose = T)) %>%
  write.csv(., "supervised-files/train-test/test.csv")
dfmat_test <- dfm_subset(dfmat, dfmat$cv_sample == 1)
save(dfmat_test, file = "supervised-files/train-test/dfmat_test.RData")
write.csv(dfmat_test$issue_r1, "supervised-files/train-test/test_val.csv")

}
```

```
# Time needed to run script (much shorter when textmodels are just loaded from a
# file)
print(Sys.time() - start_time)
```

```
## Time difference of 34.74296 secs
```