Preparing the textual data

Cornelius Erfort

9 Aug 2021

Contents

1 Setting up

This script requires the file "labeled.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",
    "xlsx")

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

set.seed(4325)
```

1.2 Loading data

For the evaluation of classifiers, we use the data for Germany.

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

```
labeled <- read.xlsx("data/labeled/germany-labeled.xlsx", sheetIndex = 1) %>%
    suppressWarnings()
```

```
# Clean
labeled <- labeled %>%
    mutate(issue = ifelse(issue == ".", NA, issue))
labeled$issue[is.na(labeled$issue)] <- labeled$X1st.coding..issue[is.na(labeled$issue)]
labeled <- labeled %>%
    mutate(issue = issue %>%
        str_replace_all(c(`19a` = "191", `19b` = "192")) %>%
        as.numeric())
```

```
## Warning in issue %>% str_replace_all(c(`19a` = "191", `19b` = "192")) %>% : NAs
## introduced by coercion
```

```
labeled$issue[labeled$id == 229] <- 191
labeled$issue[labeled$id == 983] <- 7
labeled$issue[labeled$id == 1155] <- 10

table(labeled$issue, useNA = "always")
```

```
##
##    1    2    3    4    5    6    7    8    9   10   12   13   14   15   16   17
##  175  181  119   99  167  137   84  105  131   74  195  104   32  168  121   68
##   18   20   23   98   99  191  192 <NA>
##   27   97   19   91   46  350  152  258
```

```
labeled <- filter(labeled, !is.na(issue))

labeled <- labeled %>%
    mutate(date = as.Date(as.numeric(date), origin = "1970-01-01"))
```

```
## Warning in as.Date(as.numeric(date), origin = "1970-01-01"): NAs introduced by
## coercion
```

```
# labeled <- filter(labeled, !is.na(date) | !is.na(text))
nrow(labeled)
```

```
## [1] 2742
```

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

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

```
## [1] 2742
```

```
textpress <- filter(textpress, !(header %>%
    tolower() %>%
    str_detect("einladung zum presse")))
nrow(textpress)
```

```
## [1] 2733
```

```
# Distribution of issues in the hand-coded sample
table(textpress$issue) %>%
    as.data.frame() %>%
```

issue	1	2	3	4	5	6	7	8	9	10	12	13	14	15	16	17	18	20	23	98	99	191	192
n	175	181	119	99	167	137	84	105	131	74	195	104	32	168	121	68	27	97	19	82	46	350	152

```
    dplyr::rename(issue = Var1, n = Freq) %>%
    t() %>%
    kbl(booktabs = T) %>%
    kable_styling(latex_options = "scale_down")
```

1.3 Merge 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.

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

# Merge categories
textpress <- 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
                           `98` = 99, # Non-thematic
                           `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"
issue_categories %>% dplyr::rename("Issue number" = issue_r1, "Issue name" = issue_r1_descr) %>%
  kbl(booktabs = T)
```

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	147	350	152

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")
latex_out <- issue_categories[c(1:13, 16:17, 14:15), ] %>%
  mutate(issue_r1 = as.character(issue_r1) %>% str_replace_all(c("191" = "19.1", "192" = "19.2"))) %>%
  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") %>%
  capture.output()

# Distribution with merged categories
table(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("90gruene_fraktion",
                                    "afd_bundesverband", "afd_fraktion",
                                    "fdp_bundesverband", "fdp_fraktion",
                                    "linke_fraktion", "spd_fraktion",
                                    "union_fraktion"),
                          party_name = c("Bündnis 90/Die Grünen - Fraktion",
                                         "AfD - Bundesverband", "AfD - Fraktion",
                                         "FDP - Bundesverband", "FDP - Fraktion",
                                         "DIE LINKE - Fraktion", "SPD - Fraktion",
                                         "CDU/CSU - Fraktion"))
textpress <- merge(textpress, party_names, by = "party")

# Distribution by parties
```

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

party	n
AfD - Bundesverband	111
AfD - Fraktion	11
Bündnis 90/Die Grünen - Fraktion	484
CDU/CSU - Fraktion	384
DIE LINKE - Fraktion	639
FDP - Bundesverband	231
FDP - Fraktion	298
SPD - Fraktion	575

```
table(textpress$party_name, substr(textpress$date, 1, 4)) %>%
  as.data.frame.matrix() %>% kbl(booktabs = T)
```

	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089
AfD - Bundesverband	0	0	0	4	22	9	22	23	19	12
AfD - Fraktion	0	0	0	0	0	0	0	11	0	0
Bündnis 90/Die Grünen - Fraktion	90	77	61	49	45	44	43	23	38	12
CDU/CSU - Fraktion	61	64	48	51	52	36	38	24	10	0
DIE LINKE - Fraktion	88	78	73	82	66	67	53	58	54	20
FDP - Bundesverband	13	14	12	5	45	33	41	39	19	10
FDP - Fraktion	62	66	64	47	0	0	0	6	33	20
SPD - Fraktion	112	90	95	63	54	51	36	32	30	12

```
# Combine header and text
textpress$htext <- str_c(textpress$header, " ", textpress$text)

# Make order of documents random
textpress <- textpress[sample(1:nrow(textpress), nrow(textpress)), ]
textpress$cv_sample <- sample(1:5, nrow(textpress), replace = T)

# Save dataframe
if(!file.exists("supervised-files/data/textpress.RData")) save(textpress, file = "supervised-files/data/
  load("supervised-files/data/textpress.RData")
}
```

2 Supervised and semi-supervised models

2.1 Creating the document frequency matrix (dfm)

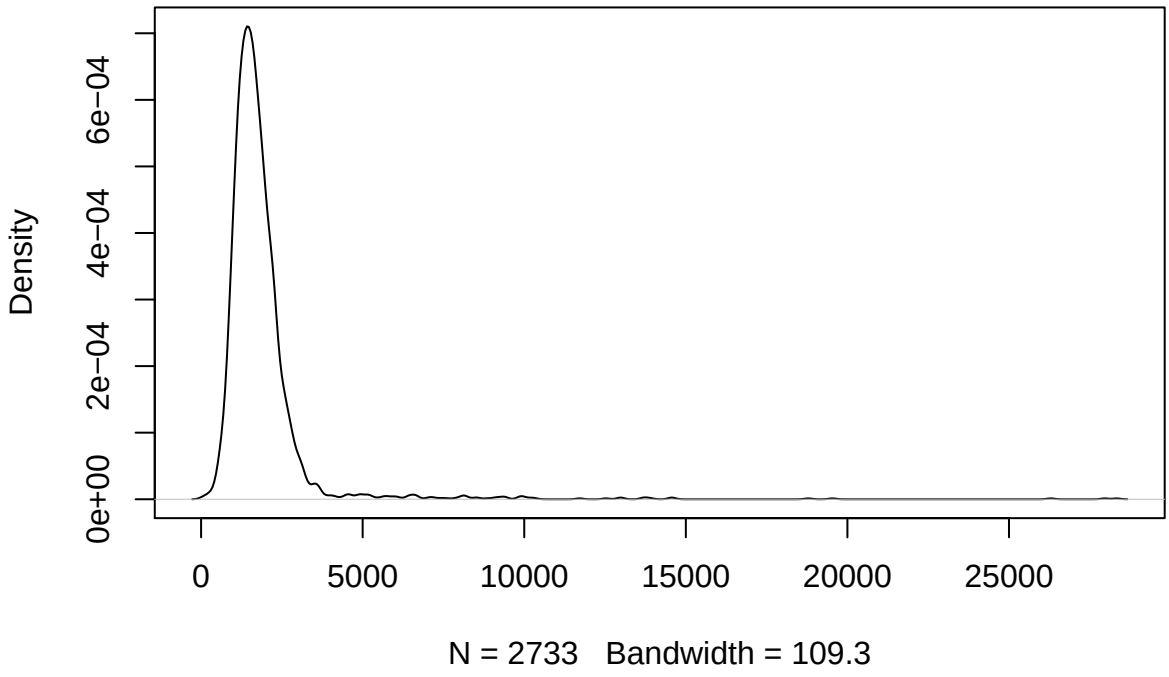
We create a text corpus based on the header and text of each press release. We draw a random sample from the corpus to create a training and a test dataset. The test dataset consists of approx. one fifth of the documents.

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.

```r
if(!dir.exists("supervised-files")) dir.create("supervised-files")

if(file.exists("supervised-files/data/dfmat.RData")) load("supervised-files/data/dfmat.RData") else {

corp_press <- corpus(str_c(textpress$header, " ", textpress$text),
                     docvars = select(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, max_docfreq = .9, # Remove words occurring <.5% or > 80% of docs
           docfreq_ = "prop") %>%
  suppressWarnings()
save(dfmat, file = "supervised-files/data/dfmat.RData")

## Create training and test set (also as csv for Python)
cbind(cv_sample = dfmat$cv_sample, label = dfmat$issue_r1, as.data.frame(dfmat)) %>% select(-c(doc_id))

# Create alternative dfm (bigrams and tfidf)
dfmat_alt <- corpus_subset(corp_press) %>%
  tokens() %>% tokens_ngrams(n = 1:2) %>%
  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(max_docfreq = .06, # Remove words occurring >6% of docs
           docfreq_ = "prop") %>%
   dfm_trim(min_docfreq = 5, # Remove words occurring in <5 docs
           docfreq_ = "count") %>% suppressWarnings()

save(dfmat_alt, file = "supervised-files/data/dfmat_alt.RData")

## Create training and test set (also as csv for Python)
cbind(cv_sample = dfmat_alt$cv_sample, label = dfmat_alt$issue_r1, as.data.frame(dfmat_alt)) %>% select
}
```

3 Transfer learning models

```r
if (!dir.exists("transfer-files")) dir.create("transfer-files")

# Load and subset data
load("supervised-files/data/textpress.RData")
textpress <- select(textpress, c(htext, issue_r1, cv_sample))

# Show distribution of text length
sapply(textpress$htext, str_length) %>%
    density() %>%
    plot
```




```
# Count words/tokens
sapply(textpress$htext, function(x) lengths(gregexpr("\\W+", x)) + 1) %>%
    max  # max_seq_length = 512
```

```
## [1] 6360
```

```
# Make labels comaptible with transformers (0-16 instead of CAP labels)
labels <- data.frame(issue_r1 = unique(textpress$issue_r1) %>%
    sort, label = c(0:16))
textpress <- merge(textpress, labels, by = "issue_r1")

# Write to csv
write_csv((textpress %>%
    select(-c(issue_r1)) %>%
    dplyr::rename(sentence1 = htext))[, c("label", "sentence1", "cv_sample")], file = "transfer-files/t
```

```
# Time needed to run script
print(Sys.time() - start_time)
```

```
## Time difference of 1.102273 mins
```