Preparing the textual data

Cornelius Erfort

9 Aug 2021

Contents

1 Setting up

This script requires the files which are 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 2,612 labeled press releases. The dataset is not uploaded on GitHub.

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	169	175	119	99	166	134	82	103	123	70	188	100	31	164	121	65	27	88	19	64	25	342	138

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

```
## [1] 2612
```

```
# Clean
labeled <- labeled %>%
    mutate(issue = issue %>%
        as.numeric())

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

```
##
##    1    2    3    4    5    6    7    8    9   10   12   13   14   15   16   17
##  169  175  119   99  166  134   82  103  123   70  188  100   31  164  121   65
##   18   20   23   98   99  191  192 <NA>
##   27   88   19   64   25  342  138    0
```

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

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

1.3 Prepare data

Add issue category names, unify parties, add variable for cross-validation.

```
# Category descriptions
issue_categories <- data.frame(issue = c(1:10, 12:18, 20, 23, 99, 191:192), issue_descr = c("Macroeconor
    "Civil Rights", "Health", "Agriculture", "Labor", "Education", "Environment",
    "Energy", "Immigration", "Transportation", "Law and Crime", "Social Welfare",
    "Housing", "Domestic Commerce", "Defense", "Technology", "Foreign Trade", "Government Operations",
    "Culture", "Other", "International Affairs", "European Integration"))
issue_categories %>%
    dplyr::rename(`Issue number` = issue, `Issue name` = issue_descr) %>%
    kbl(booktabs = T)
```

Issue number	Issue name
1	Macroeconomics
2	Civil Rights
3	Health
4	Agriculture
5	Labor
6	Education
7	Environment
8	Energy
9	Immigration
10	Transportation
12	Law and Crime
13	Social Welfare
14	Housing
15	Domestic Commerce
16	Defense
17	Technology
18	Foreign Trade
20	Government Operations
23	Culture
99	Other
191	International Affairs
192	European Integration

```
# 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	75
AfD - Fraktion	11
Bündnis 90/Die Grünen - Fraktion	482
CDU/CSU - Fraktion	381
DIE LINKE - Fraktion	639
FDP - Bundesverband	153
FDP - Fraktion	296
SPD - Fraktion	575

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

```
    kbl(booktabs = T)
```

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
AfD - Bundesverband	0	0	0	4	22	10	21	18	0	0
AfD - Fraktion	0	0	0	0	0	0	0	11	0	0
Bündnis 90/Die Grünen - Fraktion	90	77	61	49	45	44	44	22	38	12
CDU/CSU - Fraktion	62	62	46	52	51	36	38	24	10	0
DIE LINKE - Fraktion	88	78	73	82	66	67	53	58	54	20
FDP - Bundesverband	0	0	0	1	45	33	41	33	0	0
FDP - Fraktion	62	66	62	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)

if (!dir.exists("supervised-files")) dir.create("supervised-files")
if (!dir.exists("supervised-files/data")) dir.create("supervised-files/data")

# Save dataframe (do not overwrite because the cross-validation folds are saved
# here)
if (!file.exists("supervised-files/data/textpress.RData")) save(textpress, file = "supervised-files/data
    load("supervised-files/data/textpress.RData")
}
```

2 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.

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

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

corp_press <- corpus(str_c(textpress$header, " ", textpress$text),
                     docvars = select(textpress, c(id, issue, 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 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")

}
```

3 Superlearner

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

## Create training and test set (also as csv for Python)
cbind(cv_sample = dfmat$cv_sample, label = dfmat$issue, as.data.frame(dfmat)) %>%
    select(-c(doc_id)) %>%
    write.csv("supervised-files/data/dfmat.csv", row.names = F)
```

```
## Warning: 'as.data.frame.dfm' is deprecated.
## Use 'convert(x, to = "data.frame")' instead.
## See help("Deprecated")
```

```
## Create training and test set (also as csv for Python)
cbind(cv_sample = dfmat_alt$cv_sample, label = dfmat_alt$issue, as.data.frame(dfmat_alt)) %>%
    select(-c(doc_id)) %>%
    write.csv("supervised-files/data/dfmat_alt.csv", row.names = F)
```

```
## Warning: 'as.data.frame.dfm' is deprecated.
## Use 'convert(x, to = "data.frame")' instead.
## See help("Deprecated")
```

4 Readme, semi-supervised, and transfer models

Generate csv file with unlabeled and labeled documents.

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

```r
# Load all press releases
load("data/all/germany.RData")
```

```
## Warning in load("data/all/germany.RData"): strings not representable in native
## encoding will be translated to UTF-8
```

```r
alldocs <- germany %>%
    select(country, party, date, header, text, id)
nrow(alldocs)  # 44,950
```

```
## [1] 44950
```

```r
names(alldocs)
```

```
## [1] "country" "party"   "date"    "header"  "text"    "id"
```

```r
# Add labels and folds by id from textpress
load("supervised-files/data/textpress.RData")
alldocs <- merge(alldocs, select(textpress, c(id, issue, cv_sample)), by = "id",
    all = T)
nrow(alldocs)  # 44,950
```

```
## [1] 44950
```

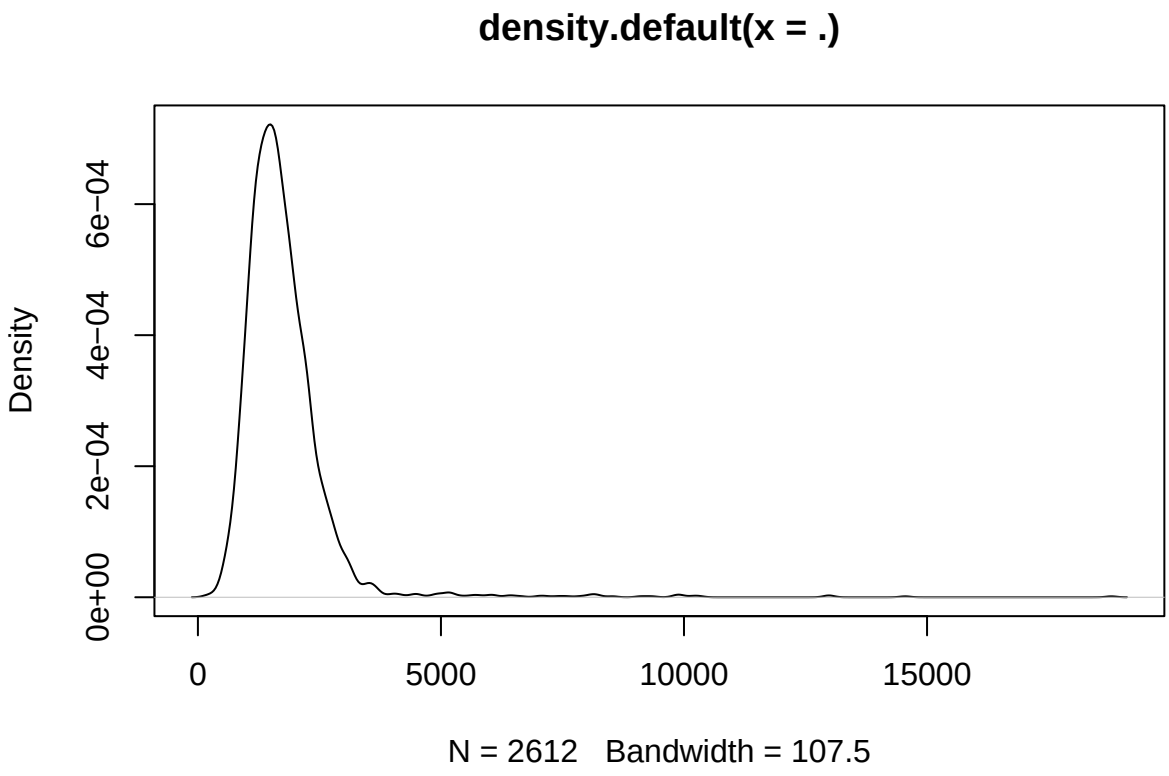
```r
alldocs$cv_sample[is.na(alldocs$cv_sample)] <- -1


# alldocs[!(alldocs$id %in% germany$id), ] %>% View

# Combine header and text
alldocs$htext <- str_c(alldocs$header, " ", alldocs$text)
alldocs <- select(alldocs, -c(header, text))

alldocs$issue[is.na(alldocs$issue)] <- -1

# Show distribution of text length (labeled data)
sapply(alldocs$htext[alldocs$issue != -1], str_length) %>%
    density() %>%
    plot
```




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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    11.0   167.0   211.0   237.8   266.0  7482.0
```

```
# Make issues compatible with transformers (0-23 instead of CAP labels)
labels <- data.frame(issue = unique(alldocs$issue) %>%
    sort, label = c(-1, 0:22))
alldocs <- merge(alldocs, labels, by = "issue", all.x = T)


table(alldocs$issue, useNA = "ifany")
```

```
##
##    -1     1     2     3     4     5     6     7     8     9    10    12    13
## 42338   169   175   119    99   166   134    82   103   123    70   188   100
##    14    15    16    17    18    20    23    98    99   191   192
##    31   164   121    65    27    88    19    64    25   342   138
```

```
# Write to csv

if (!file.exists("transfer-files/alldocs.csv")) write_csv(alldocs, file = "transfer-files/alldocs.csv")

if (!file.exists("semi-files/alldocs.csv")) write_csv(select(alldocs, -c(label)),
    file = "semi-files/alldocs.csv")
```

```r
if (!file.exists("readme-files/alldocs.RData")) select(alldocs, -c(label)) %>%
    save(., file = "readme-files/alldocs.RData")
```

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

```
## Time difference of 3.187437 mins
```