Analysis of Available Data

Load the corpora

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.2.0 --
## v broom        1.0.5     v rsample      1.2.1
## v dials        1.3.0     v tune         1.2.1
## v infer        1.0.7     v workflows    1.1.4
## v modeldata    1.4.0     v workflowsets 1.1.0
## v parsnip      1.2.1     v yardstick    1.3.2
## v recipes      1.1.0
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:scales':
```

```
##
##     alpha, rescale
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
set.seed(42)

load_kuk_subcorpus_metadata <- function(crp) {
  read_tsv(paste(c(
    "../corpora/KUK_1.0/metadata/", crp, "_DocumentFileFormat.tsv"
  ), collapse = "")) %>%
    filter(FileFormat == "TXT") %>%
    full_join(
      read_tsv(paste(c(
        "../corpora/KUK_1.0/metadata/",
        crp,
        "_DocumentIdentificationGenreProperties.tsv"
      ), collapse = "")),
      by = "KUK_ID"
    ) %>%
    mutate(across(where(is.numeric), as.character)) %>%
    mutate(subcorpus = crp) %>%
    select(KUK_ID, FileName, FileFormat, FolderPath, subcorpus, everything())
}

kuky_orig <- fromJSON("../corpora/KUKY/argumentative.json")$documents %>%
  as_tibble() %>%
  bind_rows(
    fromJSON("../corpora/KUKY/normative.json")$documents %>% as_tibble()
  ) %>%
  rename(KUK_ID = doc_id) %>%
  select(!c(plainText, doc_name)) %>%
  select(KUK_ID, everything())

kuky_kuk <- load_kuk_subcorpus_metadata("KUKY") %>%
  filter(FolderPath == "data/KUKY/TXT") %>%
  select(!c(Anonymized, RecipientType, RecipientIndividuation, AuthorType, Objectivity, LegalActType, B
```

```
## Rows: 448 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (4): KUK_ID, FileName, FileFormat, FolderPath
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 224 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (8): KUK_ID, SourceDB, Anonymized, RecipientType, RecipientIndividuation...
## lgl (4): SourceID, DocumentTitle, ClarityPursuit, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
kuky <- kuky_kuk %>% full_join(kuky_orig, by = "KUK_ID")
czcdc <- load_kuk_subcorpus_metadata("CzCDC")
```

```
## Rows: 237723 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (4): KUK_ID, FileName, FileFormat, FolderPath
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 237723 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (10): KUK_ID, SourceDB, SourceID, DocumentTitle, Anonymized, RecipientTy...
## lgl  (2): ClarityPursuit, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
eso <- load_kuk_subcorpus_metadata("ESO")
```

```
## Rows: 11230 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (3): KUK_ID, FileFormat, FolderPath
## dbl (1): FileName
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 5615 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (10): KUK_ID, SourceDB, SourceID, DocumentTitle, Anonymized, RecipientTy...
## lgl  (2): ClarityPursuit, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
frbo <- load_kuk_subcorpus_metadata("FrBo") %>%
  # load metadata for FrBo updated with Quality (=Readability)
  bind_rows(
    read_csv("../corpora/FrBo_contents.csv") %>%
      mutate(Readability = str_to_lower(Quality)) %>%
      mutate(across(c(Readability), ~ str_replace(.x, "good", "high"))) %>%
      select(!Quality)
  ) %>%
  # and move the Quality values to the original rows
  arrange(KUK_ID) %>%
  group_by(KUK_ID) %>%
  fill(Readability, .direction = "up") %>%
  ungroup() %>%
  filter(!is.na(FileName))
```

```
## Rows: 638 Columns: 4
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: "\t"
## chr (4): KUK_ID, FileName, FileFormat, FolderPath
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 319 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (10): KUK_ID, SourceDB, SourceID, DocumentTitle, Anonymized, RecipientTy...
## lgl  (2): ClarityPursuit, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 310 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): KUK_ID, SourceDB, SourceID, DocumentTitle, Quality, Anonymized, Re...
## lgl  (2): ClarityPursuit, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
lifrlaw <- load_kuk_subcorpus_metadata("LiFRLaw")
```

```
## Rows: 36 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (4): KUK_ID, FileName, FileFormat, FolderPath
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 18 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (9): KUK_ID, SourceDB, SourceID, DocumentTitle, Anonymized, Recipient Ty...
## lgl (2): ClarityPursuit, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ombuflyers <- load_kuk_subcorpus_metadata("OmbuFlyers")
```

```
## Rows: 234 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (4): KUK_ID, FileName, FileFormat, FolderPath
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 117 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (8): KUK_ID, DocumentTitle, Anonymized, RecipientType, RecipientIndividu...
## lgl (4): SourceDB, SourceID, ClarityPursuit, Bindingness
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- kuky %>%
  bind_rows(czcdc) %>%
  bind_rows(eso) %>%
  bind_rows(frbo) %>%
  bind_rows(lifrlaw) %>%
  bind_rows(ombuflyers)

str(df)
```

```
## tibble [244,016 x 21] (S3: tbl_df/tbl/data.frame)
##  $ KUK_ID               : chr [1:244016] "671918e2c6537d54ff0626db" "671918e2c6537d54ff0626dc" "671
##  $ FileName             : chr [1:244016] "orig_Certifikáty autorizovaných inspektorů" "red_Co je to
##  $ FileFormat           : chr [1:244016] "TXT" "TXT" "TXT" "TXT" ...
##  $ FolderPath           : chr [1:244016] "data/KUKY/TXT" "data/KUKY/TXT" "data/KUKY/TXT" "data/KUKY
##  $ subcorpus            : chr [1:244016] "KUKY" "KUKY" "KUKY" "KUKY" ...
##  $ SourceDB             : chr [1:244016] "SourceDB" "SourceDB" "SourceDB" "SourceDB" ...
##  $ SourceID             : chr [1:244016] NA NA NA NA ...
##  $ DocumentTitle        : chr [1:244016] NA NA NA NA ...
##  $ ClarityPursuit       : logi [1:244016] NA NA NA NA NA NA ...
##  $ Readability          : chr [1:244016] "low" "high" "low" "low" ...
##  $ SyllogismBased       : chr [1:244016] "false" "false" "false" "false" ...
##  $ DocumentVersion      : chr [1:244016] "Original" "Redesign" "Original" "Original" ...
##  $ ParentDocumentID     : chr [1:244016] NA NA NA NA ...
##  $ LegalActType         : chr [1:244016] "normative" "normative" "normative" "normative" ...
##  $ Objectivity          : chr [1:244016] "quasiobjective" "quasiobjective" "quasiobjective" "quasiob
##  $ Bindingness          : logi [1:244016] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ AuthorType           : chr [1:244016] "individual" "individual" "individual" "authority" ...
##  $ RecipientType        : chr [1:244016] "natural person" "natural person" "natural person" "natura
##  $ RecipientIndividuation: chr [1:244016] "public" "public" "public" "public" ...
##  $ Anonymized           : chr [1:244016] "No" "No" "No" "No" ...
##  $ Recipient Type       : chr [1:244016] NA NA NA NA ...
```

Properties of KUKY

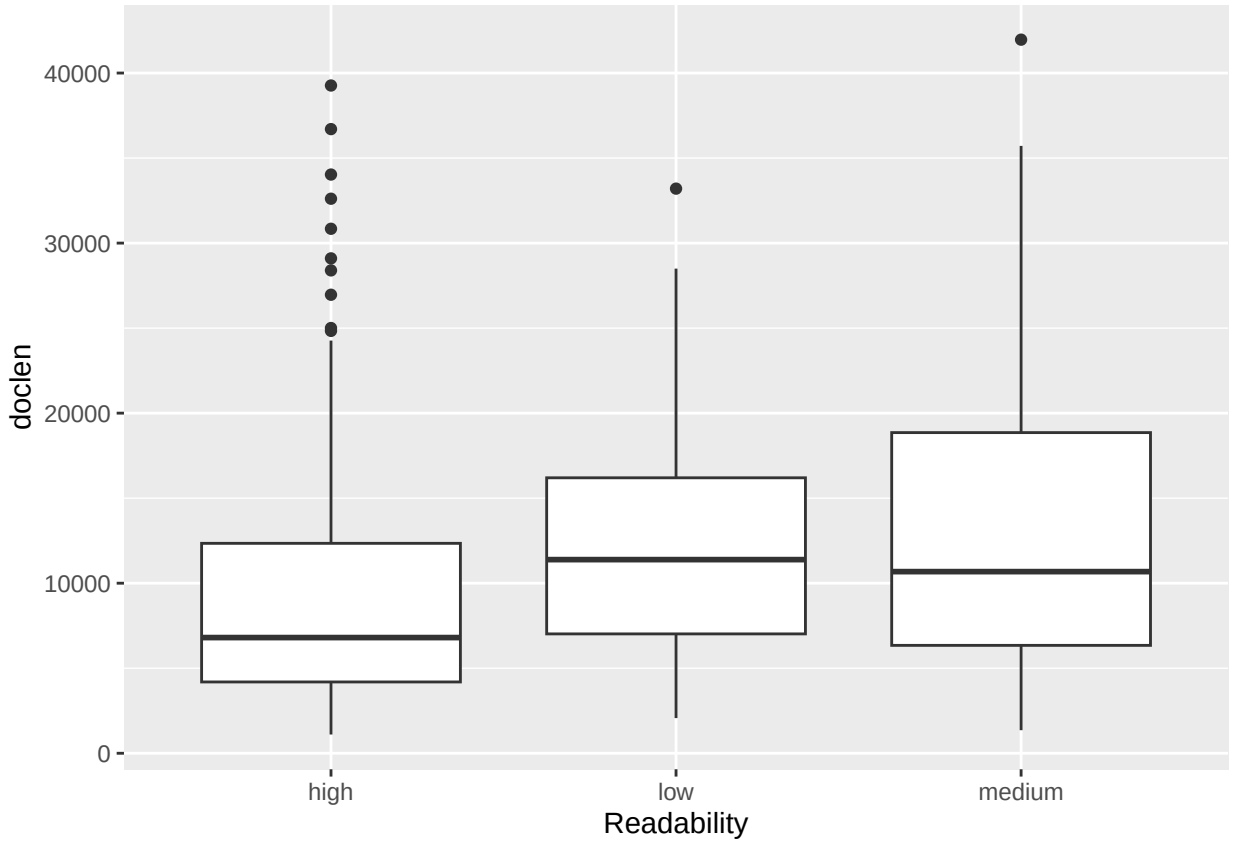
```
kuky_properties_df <- fromJSON(
  "../corpora/KUKY/argumentative.json"
)$documents %>%
  as_tibble() %>%
  bind_rows(
    fromJSON("../corpora/KUKY/normative.json")$documents %>% as_tibble()
  ) %>%
  rename(KUK_ID = doc_id) %>%
  mutate(doclen = str_length(plainText))

table(kuky_properties_df$Readability)
```

```
##
##   high    low medium
##    125     38     61
```

```
table(kuky_properties_df$Readability, kuky_properties_df$SyllogismBased)
```

```
##
##          false true
##   high      62   62
##   low       38    0
##   medium    48   11
```

```
kuky_properties_df %>% ggplot(aes(x = Readability, y = doclen)) +
  geom_boxplot()
```



Quick peek into other parts of the data set:

Subcorpus	Low # of chars	High # of chars
CzCDC/ConCo	2.000	18.000
CzCDC/SupAdmCo	3.000	30.000
CzCDC/SupCo	3.000	10.000
ESO	7.000	40.000
FrBo/articles	4.000	15.000

Properties of FrBo

```
table(frbo$FolderPath, frbo$Readability, useNA = "ifany")
```

```
##
##                           high medium <NA>
##   data/FrBo/analyses/TXT    55     17    0
##   data/FrBo/articles/TXT   184     54    9
```

Filter out duplicates

Some subcorpora overlap (*FrBo* with *ESO*, and multiple subcorpora with *KUKY*).

The usage of documents with ClarityPursuit == NA is questionable, let's exclude such documents. This effectively comes with a price of excluding the whole *ESO* subcorpus, even though some of its documents are available in *KUKY*.

The usage of documents with ClarityPursuit == TRUE is also questionable as they're not reviewed in the same manner as the documents from KUKY, yet at the same time they are less likely to be as "unreadable" as the documents with ClarityPursuit == FALSE. Such documents could very well be readable, interfering with the training process.

After filtering ClarityPursuit == NA out, the only remaining overlaps are with *KUKY*. Let's keep the documents from *KUKY* as they are associated with a more careful readability evaluation.

Additionaly, there are 3 cases where a text is assessed for readability both by *KUKY* and by *FrBo*. In 2 of these cases, the assessments don't agree: the texts are assessed "low" in *KUKY*, but "medium" by *FrBo*. This doesn't matter **under the condition** that we put them both in the same class for the training (i.e., "bad"). Let's keep the observations from *KUKY* for simplicity.

```
table(df$subcorpus, df$ClarityPursuit, useNA = "ifany")
```

```
##
##                FALSE   TRUE   <NA>
##   CzCDC       237723      0      0
##   ESO              0      0   5615
##   FrBo           114    205      0
##   KUKY             0      0    224
##   LiFRLaw          6     12      0
##   OmbuFlyers      52     65      0
```

```
table(df$ClarityPursuit, df$Readability, df$subcorpus, useNA = "ifany")
```

```
## , ,  = CzCDC
##
##
##           high    low medium   <NA>
##   FALSE      0      0      0 237723
##   TRUE       0      0      0      0
##   <NA>       0      0      0      0
##
## , ,  = ESO
##
##
##           high    low medium   <NA>
##   FALSE      0      0      0      0
##   TRUE       0      0      0      0
##   <NA>       0      0      0   5615
##
## , ,  = FrBo
##
##
##           high    low medium   <NA>
##   FALSE     51      0     54      9
##   TRUE     188      0     17      0
##   <NA>       0      0      0      0
##
```

```
## , ,  = KUKY
##
##
##           high    low medium   <NA>
##   FALSE      0      0      0      0
##   TRUE       0      0      0      0
##   <NA>     125     38     61      0
##
## , ,  = LiFRLaw
##
##
##           high    low medium   <NA>
##   FALSE      0      0      0      6
##   TRUE       0      0      0     12
##   <NA>       0      0      0      0
##
## , ,  = OmbuFlyers
##
##
##           high    low medium   <NA>
##   FALSE      0      0      0     52
##   TRUE       0      0      0     65
##   <NA>       0      0      0      0
```

```
# display duplicate file entries
df %>%
  group_by(FileName) %>%
  mutate(n = n()) %>%
  filter(n > 1) %>%
  select(FileName, subcorpus, Readability, ClarityPursuit) %>%
  arrange(FileName) %>%
  print(n = 80)
```

```
## # A tibble: 80 x 4
## # Groups:   FileName [40]
##    FileName                                subcorpus Readability ClarityPursuit
##    <chr>                                   <chr>     <chr>       <lgl>
##  1 100                                     ESO       <NA>        NA
##  2 100                                     FrBo      high        TRUE
##  3 102                                     ESO       <NA>        NA
##  4 102                                     FrBo      high        TRUE
##  5 110                                     ESO       <NA>        NA
##  6 110                                     FrBo      medium      TRUE
##  7 14                                      ESO       <NA>        NA
##  8 14                                      FrBo      high        TRUE
##  9 142                                     ESO       <NA>        NA
## 10 142                                     FrBo      medium      TRUE
## 11 148                                     ESO       <NA>        NA
## 12 148                                     FrBo      high        TRUE
## 13 152                                     ESO       <NA>        NA
## 14 152                                     FrBo      high        TRUE
## 15 154                                     ESO       <NA>        NA
## 16 154                                     FrBo      medium      TRUE
## 17 156                                     ESO       <NA>        NA
## 18 156                                     FrBo      high        TRUE
```

```
## 19 158                                          ESO       <NA>        NA
## 20 158                                          FrBo      high        TRUE
## 21 16                                           ESO       <NA>        NA
## 22 16                                           FrBo      high        TRUE
## 23 170                                          ESO       <NA>        NA
## 24 170                                          FrBo      medium      TRUE
## 25 176                                          ESO       <NA>        NA
## 26 176                                          FrBo      medium      TRUE
## 27 18                                           ESO       <NA>        NA
## 28 18                                           FrBo      high        TRUE
## 29 190                                          ESO       <NA>        NA
## 30 190                                          FrBo      high        TRUE
## 31 200                                          ESO       <NA>        NA
## 32 200                                          FrBo      high        TRUE
## 33 202                                          ESO       <NA>        NA
## 34 202                                          FrBo      high        TRUE
## 35 204                                          ESO       <NA>        NA
## 36 204                                          FrBo      high        TRUE
## 37 206                                          ESO       <NA>        NA
## 38 206                                          FrBo      high        TRUE
## 39 208                                          ESO       <NA>        NA
## 40 208                                          FrBo      high        TRUE
## 41 24                                           ESO       <NA>        NA
## 42 24                                           FrBo      high        TRUE
## 43 28                                           ESO       <NA>        NA
## 44 28                                           FrBo      medium      TRUE
## 45 30                                           ESO       <NA>        NA
## 46 30                                           FrBo      high        TRUE
## 47 42                                           ESO       <NA>        NA
## 48 42                                           FrBo      medium      TRUE
## 49 44                                           ESO       <NA>        NA
## 50 44                                           FrBo      high        TRUE
## 51 54                                           ESO       <NA>        NA
## 52 54                                           FrBo      high        TRUE
## 53 68                                           ESO       <NA>        NA
## 54 68                                           FrBo      medium      TRUE
## 55 70                                           ESO       <NA>        NA
## 56 70                                           FrBo      high        TRUE
## 57 76                                           ESO       <NA>        NA
## 58 76                                           FrBo      high        TRUE
## 59 Duchody                                      KUKY      low         NA
## 60 Duchody                                      OmbuFlye~ <NA>        FALSE
## 61 Odpadni-vody                                 KUKY      low         NA
## 62 Odpadni-vody                                 OmbuFlye~ <NA>        FALSE
## 63 ockovani-1_kusv                              KUKY      high        NA
## 64 ockovani-1_kusv                              LiFRLaw   <NA>        TRUE
## 65 ockovani-3_orig                              KUKY      low         NA
## 66 ockovani-3_orig                              LiFRLaw   <NA>        FALSE
## 67 orig_Certifikáty autorizovaných inspekt~ KUKY      low         NA
## 68 orig_Certifikáty autorizovaných inspekt~ FrBo      medium      FALSE
## 69 orig_financovani_politickych_stran           KUKY      low         NA
## 70 orig_financovani_politickych_stran           FrBo      medium      FALSE
## 71 red_Co je to územní plánování_final_při~ KUKY      high        NA
## 72 red_Co je to územní plánování_final_při~ FrBo      high        TRUE
```

```
## 73 stavarska-1_kusv                          KUKY      high        NA
## 74 stavarska-1_kusv                          LiFRLaw   <NA>        TRUE
## 75 stavarska-2_orig                          KUKY      low         NA
## 76 stavarska-2_orig                          LiFRLaw   <NA>        FALSE
## 77 zaloba-1_orig                             KUKY      medium      NA
## 78 zaloba-1_orig                             LiFRLaw   <NA>        FALSE
## 79 zaloba-2_kusv                             KUKY      high        NA
## 80 zaloba-2_kusv                             LiFRLaw   <NA>        TRUE
```

```
# search for FrBo duplicates
df_frbo_duplicates <- df %>%
  filter(str_detect(FileName, "red_|orig_")) %>%
  mutate(new_fname = str_remove(FileName, "^[0-9]{3}_")) %>%
  group_by(new_fname) %>%
  mutate(n = n()) %>%
  ungroup() %>%
  filter(n >= 2)

all_frbo_duplicates <- df_frbo_duplicates %>% pull(FileName)

df_frbo_dup_wide <- df_frbo_duplicates %>%
  select(new_fname, subcorpus, Readability, n) %>%
  distinct(new_fname, subcorpus, Readability, n) %>%
  pivot_wider(
    names_from = subcorpus,
    values_from = Readability,
    names_prefix = "Readability_"
  ) %>%
  mutate(
    class_KUKY = as.factor(if_else(Readability_KUKY == "high", "good", "bad")),
    class_FrBo = as.factor(if_else(Readability_FrBo == "high", "good", "bad"))
  )

table(
  df_frbo_dup_wide$Readability_KUKY, df_frbo_dup_wide$Readability_FrBo,
  useNA = "ifany"
)
```

```
##
##          high medium <NA>
##   high     15      0    0
##   low       7      5    1
##   medium    1      0    0
```

```
readability_agreement <- df_frbo_dup_wide %>%
  select(class_KUKY, class_FrBo) %>%
  table()
readability_agreement
```

```
##           class_FrBo
## class_KUKY bad good
##       bad    5    8
##       good   0   15
```

```r
cohen.kappa(readability_agreement)
```

```
## Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha, levels = levels,
##     w.exp = w.exp)
##
## Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
##                  lower estimate upper
## unweighted kappa  0.12      0.4  0.68
## weighted kappa    0.12      0.4  0.68
##
##  Number of subjects = 28
```

```r
# this is valid UNDER THE CONDITION that we construct the "good" class
# out of high-readability texts only
good_frbo_duplicates <- df_frbo_dup_wide %>%
  filter(
    Readability_KUKY == Readability_FrBo | (
      (Readability_KUKY == "medium" | Readability_KUKY == "low") &
        (Readability_FrBo == "medium")
    )
  ) %>%
  pull(new_fname)

bad_frbo_duplicates <- setdiff(all_frbo_duplicates, good_frbo_duplicates)

# remove FrBo/articles-originated texts from KUKY because:
#   1. they are duplicates
#   2. they are actually represented in markdown
df %>%
  filter(subcorpus == "KUKY" & str_detect(FileName, "red_|orig_")) %>%
  pull(FileName)
```

```
##  [1] "orig_Certifikáty autorizovaných inspektorů"
##  [2] "red_Co je to územní plánování_final_přidat odkaz na manuál o RP až bude"
##  [3] "orig_financovani_politickych_stran"
##  [4] "003_red_Jak dosáhnout změny dopravního značení_final"
##  [5] "015_orig_Jak komunikovat s úřady elektronicky"
##  [6] "021_red_Jak daleko od hranice pozemku musí být umístěno elektrické vedení"
##  [7] "013_orig_10 významných práv účastníka správního řízení"
##  [8] "020_red_Jak chránit vody a správně s nimi nakládat_revKZ"
##  [9] "030_orig_Co je to a jak probíhá integrované povolování_final"
## [10] "018_red_Co je to úřední deska a jak ji využít"
## [11] "012_orig_Jak chránit vody a správně s nimi nakládat_revKZ"
## [12] "010_red_Guerilla gardening, jak zahradničit na veřejném prostranství (ne)legálně_final"
## [13] "014_red_Co je to a jak probíhá integrované povolování_final"
## [14] "031_orig_Co je to EIA_final"
## [15] "023_red_Co dělat, když soused postavil černou stvabu_final, bacha na infrgafiku"
## [16] "029_orig_Certifikáty autorizovaných inspektorů"
## [17] "032_orig_Co je to úřední deska a jak ji využít"
## [18] "026_orig_Jak jedná spolek navenek"
## [19] "041_red_Hlukové limity a udělování výjimek_prefinal"
## [20] "027_orig_Jak dosáhnout odpovědnosti úředníka za škodu"
## [21] "038_orig_Co je to korupce a klientelismus"
## [22] "025_red_GDPR Jak právo chrání osobní údaje_final"
```

```
## [23] "028_orig_Co dělat, když soused postavil černou stvabu_final, bacha na infrgafiku"
## [24] "034_red_Jak dosáhnout zrušení stanoviska EIA_final"
## [25] "036_red_Dotčený vlastník - Kdo to je a jaká má v územním plánování práva_final"
## [26] "059_red_10 významných práv účastníka správního řízení"
## [27] "044_red_financovani_politickych_stran úprava 2021"
## [28] "053_orig_Guerilla gardening, jak zahradničit na veřejném prostranství (ne)legálně_final"
## [29] "051_orig_Co je to regulační plán a jak dosáhnout jeho přijetí_původní"
## [30] "049_red_CO je černá stvaba a jak ji ponat"
```

```
df <- df %>%
  filter(subcorpus != "KUKY" | !str_detect(FileName, "red_|orig_"))

# remove FrBo articles with different readability assessments by KUKY and FrBo
df <- df %>% filter(!(FileName %in% bad_frbo_duplicates))

# these two are also duplicates
df <- df %>%
  filter(!(FileName %in% c(
    "orig_Mohou spolky ve správních žalobách používat věcné argumenty_final, odkaz na soudní ochrana sp
    "red_Mohou spolky ve správních žalobách používat věcné argumenty_final, odkaz na soudní ochrana spo
  ))) %>%
  # missing in real data
  filter(FileName != "partred_Jak chránit vody a správně s nimi nakládat")

# remove OmbuFlyer-KUKY duplicates with different names
# keep the ones from KUKY
bad_of_kuky_duplicates <- df %>%
  filter(subcorpus %in% c("KUKY", "OmbuFlyers")) %>%
  mutate(new_fname = str_remove(FileName, "^[0-9]{3}_")) %>%
  group_by(new_fname) %>%
  mutate(n = n()) %>%
  ungroup() %>%
  filter(n > 1 & subcorpus == "OmbuFlyers") %>%
  select(!c(new_fname, n)) %>%
  pull(KUK_ID)
bad_of_kuky_duplicates
```

```
##  [1] "ombuflyer_0008" "ombuflyer_0012" "ombuflyer_0016" "ombuflyer_0021"
##  [5] "ombuflyer_0023" "ombuflyer_0025" "ombuflyer_0027" "ombuflyer_0028"
##  [9] "ombuflyer_0035" "ombuflyer_0036" "ombuflyer_0041" "ombuflyer_0045"
## [13] "ombuflyer_0047" "ombuflyer_0051" "ombuflyer_0054" "ombuflyer_0058"
## [17] "ombuflyer_0060" "ombuflyer_0062" "ombuflyer_0064" "ombuflyer_0105"
```

```
df <- df %>% filter(!(KUK_ID %in% bad_of_kuky_duplicates))

# keep only rows where either Readability or ClarityPursuit isn't NA
# and exclude ClarityPursuit == TRUE
df <- df %>%
  filter(!is.na(Readability) | ClarityPursuit == FALSE)

# 6 duplicates remaining
# keep the ones from KUKY as they have a readability assessment (see above)
df <- df %>%
  group_by(FileName) %>%
  mutate(n = n()) %>%
```

```
  ungroup() %>%
  filter(n == 1 | subcorpus == "KUKY") %>%
  select(!n)
```

The dataset is now free of overlaps.

Prepare for ML

Classes

```
table(df$subcorpus, df$Readability, useNA = "ifany")
```

```
##
##                high    low medium   <NA>
##   CzCDC           0      0      0 237723
##   FrBo          229      0     71      7
##   KUKY          110     24     60      0
##   LiFRLaw         0      0      0      3
##   OmbuFlyers      0      0      0     38
```

```
df <- df %>%
  mutate(class = if_else(Readability %in% c("high"), "good", "bad"))
```

Data set parameters

```
.split_prop <- 4 / 5 # proportion of testing data in the dataset
.no_folds <- 10 # no. of folds in v-fold cross-validation
.balance <- 9 / 20 # proportion of positive samples in the target dataset

dssize_positive <- count(df %>% filter(class == "good"))[[1, 1]]
dssize_total <- dssize_positive / .balance
dssize_negative <- dssize_total - dssize_positive

cat(c(
  paste(c(
    "Data set size: ", dssize_total, "\n"
  ), collapse = ""),
  paste(c(
    "Positive class size: ", dssize_positive, "\n"
  ), collapse = ""),
  paste(c(
    "Negative class size: ", dssize_negative, "\n"
  ), collapse = ""),
  paste(c(
    "Training data set size: ", dssize_total * .split_prop, "\n"
  ), collapse = ""),
  paste(c(
    "Training positive class size: ", dssize_positive * .split_prop, "\n"
  ), collapse = ""),
  paste(c(
    "Training negative class size: ", dssize_negative * .split_prop, "\n"
  ), collapse = ""),
  paste(c(
```

```
    "One fold size: ", (dssize_total * .split_prop) / .no_folds, "\n"
  ), collapse = ""),
  paste(c(
    "One fold positive class size: ", (dssize_positive * .split_prop) / .no_folds, "\n"
  ), collapse = ""),
  paste(c(
    "One fold negative class size: ", (dssize_negative * .split_prop) / .no_folds, "\n"
  ), collapse = ""),
  paste(c(
    "Evaluation data set size: ", dssize_total * (1 - .split_prop), "\n"
  ), collapse = ""),
  paste(c(
    "Evaluation positive class size: ", dssize_positive * (1 - .split_prop), "\n"
  ), collapse = ""),
  paste(c(
    "Evaluation negative class size: ", dssize_negative * (1 - .split_prop), "\n"
  ), collapse = "")
))
```

```
## Data set size: 753.333333333333
##  Positive class size: 339
##  Negative class size: 414.333333333333
##  Training data set size: 602.666666666667
##  Training positive class size: 271.2
##  Training negative class size: 331.466666666667
##  One fold size: 60.2666666666667
##  One fold positive class size: 27.12
##  One fold negative class size: 33.1466666666667
##  Evaluation data set size: 150.666666666667
##  Evaluation positive class size: 67.8
##  Evaluation negative class size: 82.8666666666667
```

Data set undersampling and split

```
table(df$subcorpus, df$class)
```

```
##
##                 bad   good
##   CzCDC      237723      0
##   FrBo           78    229
##   KUKY           84    110
##   LiFRLaw         3      0
##   OmbuFlyers     38      0
```

```
table(df$ClarityPursuit, df$class, useNA = "ifany")
```

```
##
##            bad   good
##   FALSE 237825     44
##   TRUE      17    185
##   <NA>      84    110
```

```
bads <- df %>%
  filter(class == "bad") %>%
  group_by(subcorpus) %>%
```

```
  mutate(subcorpus_size = n()) %>%
  ungroup()

max_negative_subcorpus <- bads %>%
  arrange(-subcorpus_size) %>%
  head(n = 1)

mns_name <- max_negative_subcorpus %>% pull(subcorpus)
mns_size <- max_negative_subcorpus %>% pull(subcorpus_size)
orig_negative_class_size <- bads %>%
  count() %>%
  pull(n)

# target undersample of MNS = target neg. size - other-negative-subcorpora-size
mns_target_size <- dssize_negative - (orig_negative_class_size - mns_size)

mns_sample <- sample(
  bads %>% filter(subcorpus == mns_name) %>% pull(KUK_ID), mns_target_size
)

df <- df %>% filter(
  class == "good" |
    subcorpus != mns_name |
    KUK_ID %in% mns_sample
)

table(df$subcorpus, df$class)
```

```
##
##              bad good
##   CzCDC      211    0
##   FrBo        78  229
##   KUKY        84  110
##   LiFRLaw      3    0
##   OmbuFlyers  38    0
```

```
write_csv(df, "selected_documents.csv")

# write_csv(
#   df %>%
#     select(
#       KUK_ID,
#       class,
#       FileName,
#       FolderPath,
#       subcorpus,
#       DocumentTitle,
#       Readability,
#       ClarityPursuit,
#       SyllogismBased,
#       SourceDB
#     ),
#   "selected_documents.csv"
# )
```

```r
# the split and folds aren't needed at the moment
# they'll be required in the training phase

df_split <- df %>% initial_split(prop = .split_prop)
training_set <- training(df_split)
evaluation_set <- testing(df_split)

folds <- vfold_cv(training_set, v = .no_folds, strata = class)

print(df_split)
```

```
## <Training/Testing/Total>
## <602/151/753>
```

```r
print(folds)
```

```
## #  10-fold cross-validation using stratification
## # A tibble: 10 x 2
##    splits           id
##    <list>           <chr>
##  1 <split [541/61]> Fold01
##  2 <split [541/61]> Fold02
##  3 <split [541/61]> Fold03
##  4 <split [542/60]> Fold04
##  5 <split [542/60]> Fold05
##  6 <split [542/60]> Fold06
##  7 <split [542/60]> Fold07
##  8 <split [542/60]> Fold08
##  9 <split [542/60]> Fold09
## 10 <split [543/59]> Fold10
```