

## ch6\_ex

Anna

2025-01-07

- 1) Go to the website polidoc and download manifestos of your choice in one language. Create a dfm and run a wordfish model.

Download the data of your choice and create a dfm: We chose manifestos from Austria from 2017 from polidoc. You can find those in the folder "austria\_manifestos"

```
datadir <- "./data_exercises"
```

```
library(readtext)
```

```
library(quanteda) #read in necessary packages
```

```
## Warning: Paket 'quanteda' wurde unter R Version 4.2.3 erstellt
```

```
## Package version: 3.3.1
```

```
## Unicode version: 13.0
```

```
## ICU version: 69.1
```

```
## Parallel computing: 12 of 12 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
manifestos_df<- readtext(paste0(datadir,"/austria_manifestos"), encoding= "UTF-8") #read in data from f
```

```
country <- c("Austria", "Austria", "Austria", "Austria", "Austria") #add a country variable
```

```
party <- c("Die Grünen", "KPÖ", "SPÖ", "FPÖ", "NEOS") #party of manifesto variable - same order as in fo
```

```
year <- c("2017", "2017", "2017", "2017", "2017") #year of manifesto variable
```

```
manifestos_df$country <- country
```

```
manifestos_df$year <- year
```

```
manifestos_df$party <- party #add additional variables to data frame
```

```
corpus_manifestos <- corpus(manifestos_df) #turn dataframe into corpus
```

```
summary(corpus_manifestos) #print corpus summary
```

```
## Corpus consisting of 5 documents, showing 5 documents:
```

```
##
```

##		Text	Types	Tokens	Sentences	country	year	party
##	42110.000.2017.1.1.txt		5707	21000	677	Austria	2017	Die Grünen
##	42220.000.2017.1.1.txt		1605	4376	181	Austria	2017	KPÖ
##	42320.000.2017.1.1.txt		10076	57291	2958	Austria	2017	SPÖ
##	42420.000.2017.1.1.txt		2786	8601	289	Austria	2017	FPÖ
##	42450.000.2017.1.1.txt		8856	37117	1649	Austria	2017	NEOS

```
dfm_aus <-
```

```
  quanteda::dfm(corpus_manifestos %>%
```

```
    quanteda::tokens(
```

```
      remove_punct = TRUE,
```

```
      remove_numbers = TRUE,
```

```
      remove_symbols = TRUE,
```

```

remove_url = TRUE)) %>%
  quantda::dfm_remove(stopwords("de")) %>%
  quantda::dfm_wordstem(language = "de")
head(dfm_aus)

```

```

## Document-feature matrix of: 5 documents, 13,193 features (68.79% sparse) and 3 docvars.
##
## docs
##      features
##      grun wahlprogramm nationalratswahl lieb leserin les
## 42110.000.2017.1.1.txt      70          3          1      2          1      1
## 42220.000.2017.1.1.txt       0          3          2      1          0      0
## 42320.000.2017.1.1.txt       0          0          1      7          0      6
## 42420.000.2017.1.1.txt       1          2          1      0          0      0
## 42450.000.2017.1.1.txt       4          0          0      2          0      2
##
##      features
## docs
##      osterreich europa steh gross
## 42110.000.2017.1.1.txt      112      46      12      30
## 42220.000.2017.1.1.txt       16       2       2       5
## 42320.000.2017.1.1.txt      322      69      28      64
## 42420.000.2017.1.1.txt      112      15       7       5
## 42450.000.2017.1.1.txt      196     108      21      33
## [ reached max_nfeat ... 13,183 more features ]

```

Run Wordfish: We need to rank two texts, we make the justifiable statement that Die Grünen are left of FPÖ.

```

library(quantda.textmodels)
library(quantda.textplots)
fish_aus <- textmodel_wordfish(dfm_aus, dir = c(1,4)) #1 = Die Grünen, 4 = FPÖ
summary(fish_aus, n = 5)

```

```

##
## Call:
## textmodel_wordfish.dfm(x = dfm_aus, dir = c(1, 4))
##
## Estimated Document Positions:
##      theta      se
## 42110.000.2017.1.1.txt 0.006585 0.017813
## 42220.000.2017.1.1.txt 0.061448 0.039101
## 42320.000.2017.1.1.txt -1.611993 0.009437
## 42420.000.2017.1.1.txt 0.451237 0.022944
## 42450.000.2017.1.1.txt 1.092724 0.007745
##
## Estimated Feature Scores:
##      grun wahlprogramm nationalratswahl lieb leserin
## beta 0.1791      0.2392      -0.04765 -0.4833 0.1127
## psi  2.4639      0.2198      -0.25955 0.4065 -1.8517

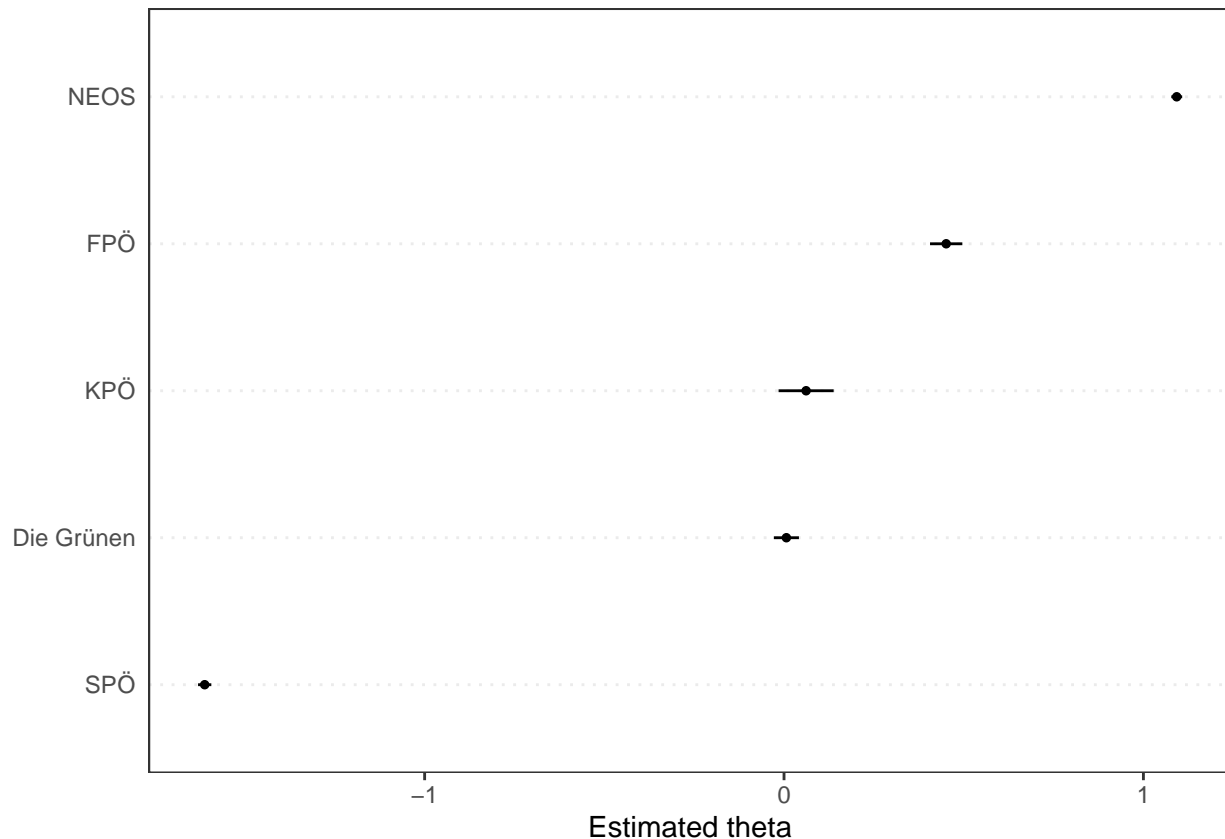
```

You can visualize the results:

```

textplot_scale1d(fish_aus, doclabels=fish_aus$x@docvars$party)

```



- 2) Replicate the code in the k-means section for the word2vec model on the data used here, or choose another text dataset of your choice. You will keep using this model for Exercise 2 of Chapter 7.

To correct your replication you can check the chapter again, we do it here quickly:

```
load(paste0(datadir, "\\supercorpus_unstemmed_V4.RData")) #load where you saved it
library(word2vec)
```

```
## Warning: Paket 'word2vec' wurde unter R Version 4.2.3 erstellt
```

```
clean_doc <- txt_clean_word2vec(corpus_unstemmed[["documents"]][["texts"]], ascii = TRUE, alpha = TRUE,
library(stopwords)
set.seed(45) #same results every run
model_w2v <- word2vec(clean_doc, stopwords = stopwords("en")) #run model
```

- a) Find out what the six nearest neighbours, meaning most similar words, are to the word “ideology”.

```
set.seed(5)
predict(model_w2v, "ideology", type = "nearest", top_n = 6)
```

```
## $ideology
##      term1      term2 similarity rank
## 1 ideology sectarianism 0.9105780 1
## 2 ideology ideologies 0.8891307 2
## 3 ideology fanaticism 0.8814456 3
## 4 ideology intolerant 0.8794775 4
## 5 ideology islamism 0.8697965 5
## 6 ideology isolationism 0.8632846 6
```

In this data intolerant, dogmatism, obsession and fanaticism are closely related to ideology.

b) Choose your own word and find the words closest to it based on the speeches dataset.

```
set.seed(5)
predict(model_w2v, "racism", type = "nearest", top_n = 1)
```

```
## $racism
##      term1      term2 similarity rank
## 1 racism xenophobia 0.9435748     1
```

For racism the nearest neighbor is xenophobia.

c) Find the embedding vector for the word “speech”. Hint: word2vec has documentation that you can access online.

The documentation on CRAN shows that we can use the predict function again:

```
??word2vec::predict
```

```
## starte den http Server für die Hilfe fertig
```

```
set.seed(5)
predict(model_w2v, "speech", type = "embedding")
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## speech -0.4132401 0.2589686 1.234186 -0.7934176 0.2914565 1.943635 -0.2320847
##           [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## speech -0.3850922 -0.703484 -2.018492 0.5110597 -0.2108572 0.129309 0.1394332
##           [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]
## speech 0.9234478 -1.622347 1.092494 -1.357701 0.5006937 -1.085609 0.3982497
##           [,22]     [,23]     [,24]     [,25]     [,26]     [,27]     [,28]
## speech -1.076228 2.001268 -0.5929797 -2.207461 0.4211437 -0.2440408 0.9309615
##           [,29]     [,30]     [,31]     [,32]     [,33]     [,34]     [,35]
## speech -1.232476 1.249522 -1.75277 0.7486828 -1.059155 1.327543 -0.946647
##           [,36]     [,37]     [,38]     [,39]     [,40]     [,41]     [,42]
## speech 2.12924 -0.8320267 -0.368935 -0.1322898 -0.4644123 0.2464057 0.3786531
##           [,43]     [,44]     [,45]     [,46]     [,47]     [,48]
## speech -0.1121604 -1.085584 0.8099266 -0.1475246 -0.5356416 -0.4181107
##           [,49]     [,50]
## speech -0.6182843 0.164831
```

3) Use the data from Exercise 1. Replicate the LDA analysis from this chapter, finding out how much the chosen parties talk about renewable energy and unemployment. You will keep using this model in Exercise 3 of Chapter 7.

In exercise 1, we worked on the whole document, here we are interested in individual sentences, therefore we need to reshape the corpus and then create a dtm:

```
corpus_sent <- corpus_reshape(corpus_manifestos, to = "sentences") #transform into sentence corpus

dfm_sent_aus <-
  quanteda::dfm(corpus_sent %>%
    quanteda::tokens(
      remove_punct = TRUE,
      remove_numbers = TRUE,
      remove_symbols = TRUE,
      remove_url = TRUE)) %>%
  quanteda::dfm_remove(stopwords("de")) %>%
```

```
quanteda::dfm_wordstem(language = "de")
head(dfm_sent_aus)
```

```
## Document-feature matrix of: 6 documents, 13,193 features (99.93% sparse) and 3 docvars.
##               features
## docs          grun wahlprogramm nationalratswahl lieb leserin les
## 42110.000.2017.1.1.txt.1      1          0          0  0          0  0
## 42110.000.2017.1.1.txt.2      1          1          1  2          1  1
## 42110.000.2017.1.1.txt.3      0          0          0  0          0  0
## 42110.000.2017.1.1.txt.4      0          0          0  0          0  0
## 42110.000.2017.1.1.txt.5      0          0          0  0          0  0
## 42110.000.2017.1.1.txt.6      1          0          0  0          0  0
##               features
## docs          osterreich europa steh gross
## 42110.000.2017.1.1.txt.1          0      0  0      0
## 42110.000.2017.1.1.txt.2          0      0  0      0
## 42110.000.2017.1.1.txt.3          1      1  1      1
## 42110.000.2017.1.1.txt.4          0      0  0      0
## 42110.000.2017.1.1.txt.5          0      1  0      0
## 42110.000.2017.1.1.txt.6          0      0  0      0
## [ reached max_nfeat ... 13,183 more features ]
```

For LDA we need a DTM:

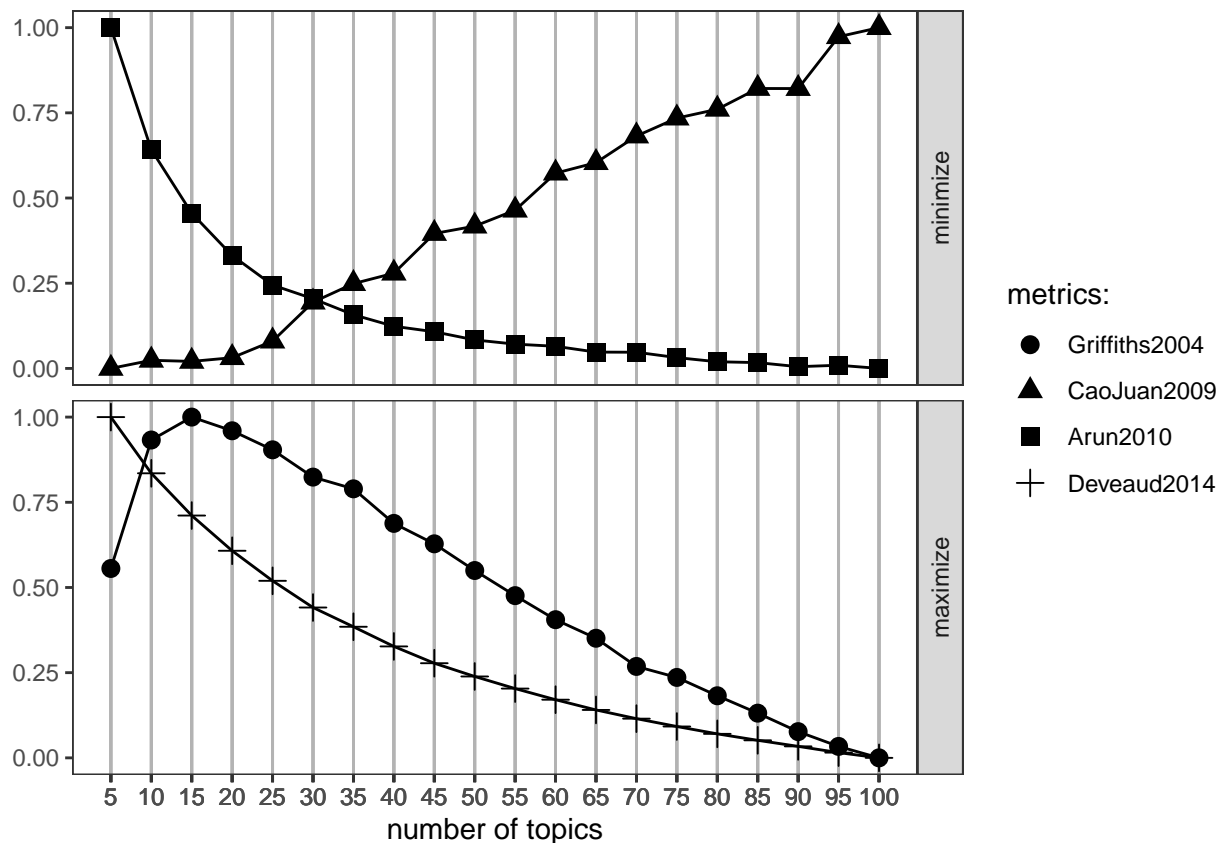
```
dtm_aus <- convert(dfm_sent_aus, to = "topicmodels")
```

Finding k: We use the ldatuning package to decide on k.

```
library("ldatuning") #install this package if you haven't yet, then use library
result_aus <- FindTopicsNumber(
  dtm_aus, #pass dtm
  topics = seq(from = 5, to = 100, by = 5), #calculated amount of topics
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010", "Deveaud2014"), #the algorithms
  method = "Gibbs", #the sampling algorithm
  control = list(seed = 77), #same result if code is run again
  verbose = TRUE #so R tells you what it is working on right now
)
```

```
## fit models... done.
## calculate metrics:
## Griffiths2004... done.
## CaoJuan2009... done.
## Arun2010... done.
## Deveaud2014... done.
```

```
FindTopicsNumber_plot(result_aus)
```



Looking at the maximum of Griffiths, I choose 15 here.

Fit the LDA model: We use the topicmodels package and the k from above.

```
library(topicmodels) #load
```

```
## Warning: Paket 'topicmodels' wurde unter R Version 4.2.3 erstellt
```

```
set.seed(50) #same result if code is run again
```

```
model_aus <- topicmodels::LDA(dtm_aus, k = 15, method = "Gibbs", control = list(alpha = 0.5, burnin = 10000))
```

Let's look at the terms to find renewable energy and unemployment:

```
terms(model_aus, 20)
```

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
## [1,]	"kind"	"uns"	"massnahm"	"euro"	"offent"
## [2,]	"ausbau"	"gesellschaft"	"ziel"	"prozent"	"mehr"
## [3,]	"bess"	"herausforder"	"klar"	"jahr"	"sowi"
## [4,]	"bereich"	"sich"	"plan"	"osterreich"	"investition"
## [5,]	"recht"	"muss"	"dah"	"pro"	"forder"
## [6,]	"famili"	"land"	"osterreich"	"rund"	"hoh"
## [7,]	"pfleg"	"beitrag"	"braucht"	"derzeit"	"privat"
## [8,]	"zugang"	"gross"	"verantwort"	"million"	"sozial"
## [9,]	"ab"	"zukunft"	"a"	"jahrlich"	"ausbau"
## [10,]	"unabhang"	"wichtig"	"konkret"	"monat"	"neu"
## [11,]	"ermog"	"stell"	"bring"	"zwei"	"wohn"
## [12,]	"lang"	"sozial"	"setz"	"milliard"	"geld"
## [13,]	"etc"	"leist"	"umsetz"	"mehr"	"energ"

##	[14,]	"finanziell"	"gerecht"	"kunst-"	"drei"	"leistbar"
##	[15,]	"versorg"	"wirtschaft"	"gut"	"davon"	"langfrist"
##	[16,]	"angebot"	"zeit"	"kultur"	"hoh"	"infrastruktur"
##	[17,]	"sowi"	"interess"	"kunst"	"liegt"	"einfuhr"
##	[18,]	"betreu"	"wohlstand"	"form"	"steu"	"mittel"
##	[19,]	"ausreich"	"gemeinsam"	"notwend"	"kost"	"arbeitsplatz"
##	[20,]	"gleichzeit"	"fair"	"land"	"etwa"	"erneuerbar"
##		Topic 6	Topic 7	Topic 8	Topic 9	
##	[1,]	"dass"	"europa"	"entwickl"	"mensch"	
##	[2,]	"osterreich"	"polit"	"forschung"	"leb"	
##	[3,]	"uns"	"stark"	"bereich"	"alt"	
##	[4,]	"bereich"	"osterreich"	"international"	"viel"	
##	[5,]	"imm"	"staat"	"schaff"	"gut"	
##	[6,]	"dafur"	"eu"	"osterreich"	"aufgrund"	
##	[7,]	"dah"	"gemeinsam"	"neu"	"recht"	
##	[8,]	"sorg"	"aufgab"	"universitat"	"jung"	
##	[9,]	"schutz"	"demokrati"	"unternehmen"	"person"	
##	[10,]	"vier"	"aktiv"	"wirtschaft"	"oft"	
##	[11,]	"muss"	"statt"	"hochschul"	"gesellschaft"	
##	[12,]	"fairness"	"braucht"	"unterstutz"	"mehr"	
##	[13,]	"beseit"	"menschenrecht"	"rahmenbeding"	"imm"	
##	[14,]	"verdi"	"union"	"bereit"	"eig"	
##	[15,]	"unfair"	"eben"	"studier"	"freiheit"	
##	[16,]	"gegenub"	"militar"	"stark"	"behinder"	
##	[17,]	"staat"	"sich"	"insbesond"	"seh"	
##	[18,]	"weiterhin"	"zusammenarbeit"	"innovation"	"darf"	
##	[19,]	"sich"	"sinn"	"forder"	"gerad"	
##	[20,]	"aufgab"	"neu"	"wettbewerb"	"folg"	
##		Topic 10	Topic 11	Topic 12	Topic 13	
##	[1,]	"osterreich"	"schul"	"muss"	"moglich"	
##	[2,]	"nachhalt"	"soll"	"gesetz"	"soll"	
##	[3,]	"erhalt"	"ausbild"	"verpflicht"	"land"	
##	[4,]	"wirtschaft"	"mehr"	"moglich"	"entsprech"	
##	[5,]	"landwirtschaft"	"chanc"	"mehr"	"einfach"	
##	[6,]	"besond"	"neu"	"dass"	"gibt"	
##	[7,]	"raum"	"bildung"	"gilt"	"transparent"	
##	[8,]	"umwelt"	"kind"	"unternehmen"	"bund"	
##	[9,]	"natur"	"digital"	"anspruch"	"einzeln"	
##	[10,]	"gesund"	"lehrling"	"regel"	"offent"	
##	[11,]	"klein"	"schulerinn"	"bzw"	"gemeind"	
##	[12,]	"produkt"	"betrieb"	"kommt"	"zustand"	
##	[13,]	"regional"	"den"	"arbeitnehmerinn"	"derzeit"	
##	[14,]	"forder"	"ford"	"darub"	"b"	
##	[15,]	"einsatz"	"dabei"	"soll"	"neu"	
##	[16,]	"landlich"	"gefordert"	"arbeitszeit"	"steht"	
##	[17,]	"ressourc"	"kindergart"	"durf"	"verwalt"	
##	[18,]	"konzern"	"zeit"	"hinaus"	"wesent"	
##	[19,]	"international"	"dafur"	"steh"	"verfug"	
##	[20,]	"ford"	"gemeinsam"	"schnell"	"zukunft"	
##		Topic 14	Topic 15			
##	[1,]	"jahr"	"frau"			
##	[2,]	"seit"	"arbeit"			
##	[3,]	"wurd"	"wenig"			
##	[4,]	"geht"	"gleich"			

```
## [5,] "grun"      "viel"
## [6,] "osterreich" "imm"
## [7,] "schon"     "gibt"
## [8,] "bereit"    "leistung"
## [9,] "letzt"     "einkomm"
## [10,] "viel"     "mann"
## [11,] "unterschied" "dass"
## [12,] "stark"    "gering"
## [13,] "weit"     "sozial"
## [14,] "erst"     "arbeitslos"
## [15,] "land"     "lang"
## [16,] "gross"    "arbeitsmarkt"
## [17,] "mehr"     "betroff"
## [18,] "macht"    "tativ"
## [19,] "heut"     "hoch"
## [20,] "wien"     "lohn"
```

Topic 15 contains “arbeitslos” meaning unemployed. Topic 10 discusses nature but renewable energy is not highly influential on the topics. You can go deeper into the sentences and their topics to find more information:

```
a_doc_topic <- data.frame(topics(model_aus))
head(a_doc_topic, 5) #just showing the beginning of the table
```

```
##               topics.model_aus.
## 42110.000.2017.1.1.txt.1         14
## 42110.000.2017.1.1.txt.2         12
## 42110.000.2017.1.1.txt.3          2
## 42110.000.2017.1.1.txt.4          2
## 42110.000.2017.1.1.txt.5          7
```

We use the corpus on sentence basis to get a dataframe and merge it with the topics:

```
df_aus_sent <- convert(corpus_sent, to = "data.frame")

a_doc_topic$doc_id <- rownames(a_doc_topic)
aus_topics <- merge(df_aus_sent, a_doc_topic, by= "doc_id")
View(aus_topics)

env_topic <- aus_topics[which(aus_topics$topics.model_aus=="10"),]
head(env_topic$text)
```

```
## [1] "Die Schwerpunkte der Sicherheitspolitik tragen dem täglichen Risiko, im Verkehr zu schaden zu k
## [2] "Nicht umgekehrt."
## [3] "Tätigkeiten (Kfz- und Flugverkehr) sollen höher als saubere Energien besteuert werden, weil dam
## [4] "Der sogenannte Regress wird häufig durch rechtzeitiges weitergeben von Vermögen umgangen. -
## [5] "Mit dem Beschluss des Pariser Klimaabkommens im Dezember 2015 ist klar: Die Dekarbonisierung un
## [6] "Besonders wenig geht insbesondere in Österreich beim Problemsektor Nr. 1, dem Verkehr, weiter."
```

```
unemp_topic <- aus_topics[which(aus_topics$topics.model_aus=="15"),]
head(unemp_topic$text)
```

```
## [1] "Von der tarifreform 2015/16 haben Besserverdienende viel stärker profitiert als Niedrigverdiener
## [2] "Nach wie vor sind Frauen zahlreichen Diskriminierungen am Arbeitsmarkt ausgesetzt."
## [3] "Die Arbeit von Frauen wird grundsätzlich minderbewertet, zudem arbeiten Frauen sehr oft in Teil
## [4] "Niedrige und mittlere Einkommen sollen profitieren."
## [5] "Für viele wohnungssuchende sind „marktkonforme“ Mieten kaum mehr leistbar, gerade in den Ballun
## [6] "Von Arbeit leben zu können, wird häufig immer schwieriger."
```



A closer look into the sentences shows again that it is difficult to expect a specific topic, here, general environmental and work and work discrimination related issues are discussed.

Now, we want to know out how much the different parties talk about renewable energy and unemployment. We check the length of the topics with `nrow` and the length of the whole manifesto with the `sentences` variable in the corpus:

```
summary(corpus_manifestos) #check length
```

```
## Corpus consisting of 5 documents, showing 5 documents:
```

```
##
##           Text Types Tokens Sentences country year      party
## 42110.000.2017.1.1.txt  5707  21000      677 Austria 2017 Die Grünen
## 42220.000.2017.1.1.txt  1605   4376      181 Austria 2017      KPÖ
## 42320.000.2017.1.1.txt 10076  57291     2958 Austria 2017      SPÖ
## 42420.000.2017.1.1.txt  2786   8601      289 Austria 2017      FPÖ
## 42450.000.2017.1.1.txt  8856  37117     1649 Austria 2017      NEOS
```

```
(nrow(env_topic[ env_topic$party == "Die Grünen",])/677)*100
```

```
## [1] 7.090103
```

```
(nrow(env_topic[ env_topic$party == "KPÖ",])/181)*100
```

```
## [1] 3.867403
```

```
(nrow(env_topic[ env_topic$party == "SPÖ",])/2958)*100
```

```
## [1] 3.076403
```

```
(nrow(env_topic[ env_topic$party == "FPÖ",])/289)*100
```

```
## [1] 6.920415
```

```
(nrow(env_topic[env_topic$party == "NEOS",])/1649)*100
```

```
## [1] 6.852638
```

As expected, in relation to the manifesto length, the environment/renewable energy topic is most discussed by the green party followed by FPÖ and NEOS.

```
(nrow(unemp_topic[unemp_topic$party == "Die Grünen",])/677)*100
```

```
## [1] 4.874446
```

```
(nrow(unemp_topic[unemp_topic$party == "KPÖ",])/181)*100
```

```
## [1] 5.524862
```

```
(nrow(unemp_topic[unemp_topic$party == "SPÖ",])/2958)*100
```

```
## [1] 5.30764
```

```
(nrow(unemp_topic[ unemp_topic$party == "FPÖ",])/289)*100
```

```
## [1] 4.49827
```

```
(nrow(unemp_topic[unemp_topic$party == "NEOS",])/1649)*100
```

```
## [1] 4.487568
```

The unemployment topic is most discussed by KPÖ and SPÖ.