Assignment 4: Topic Modelling

Carlo Greß

2023-11-12

0. Dependencies

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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 error
```

```r
library(quanteda)
```

```
## Package version: 3.3.1
## Unicode version: 14.0
## ICU version: 71.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
```

```r
library(manifestoR)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following objects are masked from 'package:quanteda':
##
##     meta, meta<-
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
##
## Loading required package: tm
```

```
##
## Attaching package: 'tm'
##
## The following object is masked from 'package:quanteda':
##
##     stopwords
##
## When publishing work using the Manifesto Corpus, please make sure to cite it correctly and to give th
##
## You can print citation and version information with the function mp_cite().
##
## Note that some of the scaling/analysis algorithms provided with this package were conceptually devel
```

```
library(readr)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

1. Data acquisition, description, and preparation

Loading the data. I chose to work with the programs of all German parties that have been part of the German Bundestag after the Federal election 1998 and 2002. In 1998, and after 16 years, the governing coalition switched from a liberal-conservative (FDP and CDU/CSU) to a social democratic/green government (SPD/Grüne). My goal is to identify whether programmatic similarities from the respective manifestos of SPD and Grüne could potentially have predicted these transitions. For this purpose, my approach is three-fold: First, I will examine all manifestos that have been published shortly before the 1998 election to see what the dominant topics across all parties are. Second, I will take a closer look to the 1998 manifestos of SPD and Grüne to determine whether these two parties shared programmatic similarities that might explain their later collaboration in government. Lastly, I will also look at the 2002 SPD and Grüne manifestos in order to evaluate whether the parties most prevalent topics start to differ, since in 2005, the government switched again to a CDU/CSU-led government.

I am directly accessing the API and loading the manifesto corpus by restricting the country to Germany and setting the time span accordingly. As a result, only the manifestos published prior to the the 1998 and 2002 Federal Elections in Germany are considered.

Printing the my_corpus document already reveals that we are dealing with 10 distinct documents (here: 10 distinct party manifestos). This intuitively makes sense since there are five different parties that have been elected to the German Bundestag in 1998, and we are looking at two distinct elections, so there are two manifestos per party (== 10 in total).

```
mp_setapikey("manifesto_apikey.txt")

my_corpus <- mp_corpus(countryname == "Germany" & edate > as.Date("1998-01-01") & edate < as.Date("2003
```

```
## Connecting to Manifesto Project DB API...
## Connecting to Manifesto Project DB API... corpus version: 2023-1
## Connecting to Manifesto Project DB API...
## Connecting to Manifesto Project DB API... corpus version: 2023-1
## Connecting to Manifesto Project DB API... corpus version: 2023-1
## Connecting to Manifesto Project DB API... corpus version: 2023-1
```

```r
my_corpus
```

```
## <<ManifestoCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 10
```

Let's double-check which parties and years are represented in our corpus: As we can see from the output, we indeed have five distinct parties (represented by party codes) and two unique points in time. The time codes are easy to interpret: All documents are either from September 1998 or September 2002. Since Federal Elections usually take place in September and we want to analyse exactly the elections in the given years, this seems appropriate.

```r
unique(meta(my_corpus, "date"))
```

```
## [[1]]
## [1] 199809
##
## [[2]]
## [1] 200209
```

```r
unique(meta(my_corpus, "party"))
```

```
## [[1]]
## [1] 41113
##
## [[2]]
## [1] 41221
##
## [[3]]
## [1] 41320
##
## [[4]]
## [1] 41420
##
## [[5]]
## [1] 41521
```

The party codes do not make intuitive sense. Let's store them and then use the larger main data set to print the party names that are attached to these codes (these are by default not included when directly importing single documents via the API, hence we refer to the main data):

```r
party_id <- as.numeric(unique(meta(my_corpus, "party")))

main <- mp_maindataset()

main %>%
  select(party, partyname) %>%
  filter(party %in% party_id) %>%
  distinct()
```

```
## # A tibble: 5 x 2
##   party partyname
##   <dbl> <chr>
## 1 41320 Social Democratic Party of Germany
## 2 41420 Free Democratic Party
## 3 41521 Christian Democratic Union/Christian Social Union
## 4 41221 Party of Democratic Socialism
## 5 41113 Alliance‘90/Greens
```

As we can see, five distinct parties are included: The SPD, the FDP, the CDU/CSU, the PDS (now: Die Linke), and Bündnis90/Die Grünen. That matches our expectation!

As a last step before diving deeper into the analysis, let’s briefly examine how many quasi-sentences there are in each of the 10 manifestos (these are the manually annoated sentences from the manifestos). For this purpose, I am looping over the whole my_corpus object, assign the number of lines to the object num_lines, and printing the result.

```
for (i in seq_along(my_corpus)) {
  doc <- my_corpus[[i]]
  num_lines <- length(unlist(strsplit(as.character(doc), "\n")))
  cat("Manifesto", i, "consists of", num_lines, "quasi-sentences\n")
}
```

```
## Manifesto 1 consists of 2354 quasi-sentences
## Manifesto 2 consists of 1774 quasi-sentences
## Manifesto 3 consists of 1006 quasi-sentences
## Manifesto 4 consists of 892 quasi-sentences
## Manifesto 5 consists of 1143 quasi-sentences
## Manifesto 6 consists of 1770 quasi-sentences
## Manifesto 7 consists of 1745 quasi-sentences
## Manifesto 8 consists of 2118 quasi-sentences
## Manifesto 9 consists of 596 quasi-sentences
## Manifesto 10 consists of 1390 quasi-sentences
```

As we can see, the length of the 10 manifestos ranges from 596 to 2354 quasi-sentences. In total, the entire corpus contains 14788 quasi-sentences.

Next, I am already transforming the preliminary corpus into a format that the `quanteda`-package can easily use. I am keeping all the meta information since we want to make conclusions based on distinct years and parties later on. I chose to already group by year here, because I couldn’t figure out how to group for two variables (year and party) during preprocessing later. Printing the resulting objects reveals that the combined manifestos from 2002 contain more quasi-sentences than in 1998.

```
quanteda_corpus <- my_corpus %>%
  as.data.frame(with.meta = TRUE) %>%
  corpus(docid_field = "manifesto_id", unique_docnames = F)

corpus1998 <- quanteda_corpus %>% corpus_subset(date == 199809)
corpus2002 <- quanteda_corpus %>% corpus_subset(date == 200209)

corpus1998
```

```
## Corpus consisting of 6,844 documents and 18 docvars.
## 41113_199809.1 :
## "BÜNDNIS 90/DIE GRÜNEN"
##
## 41113_199809.2 :
## "Grün ist der Wechsel"
##
## 41113_199809.3 :
## "Programm zur Bundestagswahl 1998"
##
## 41113_199809.4 :
## "Inhalt"
##
## 41113_199809.5 :
## "Umwelt- und Naturschutz -"
##
## 41113_199809.6 :
## "Basis der ökologischen Wirtschaft 7"
##
## [ reached max_ndoc ... 6,838 more documents ]
```

```
corpus2002
```

```
## Corpus consisting of 7,944 documents and 18 docvars.
## 41113_200209.1 :
## "Grün wirkt!"
##
## 41113_200209.2 :
## "Unser Wahlprogramm"
##
## 41113_200209.3 :
## "2002 - 2006"
##
## 41113_200209.4 :
## "Präambel"
##
## 41113_200209.5 :
## "BÜNDNIS 90/DIE GRÜNEN sind die Partei der ökologischen Moder..."
##
## 41113_200209.6 :
## "der sozialen und wirtschaftlichen Erneuerung"
##
## [ reached max_ndoc ... 7,938 more documents ]
```

Before running the topic models, we will apply some preprocessing to the corpora for both 1998 and 2002 in order to draw more meaningful conclusions from the analysis. In the following code chunk, I am removing punctuation, numbers and German stopwords, as well as stemming the whole corpus. Furthermore, I am excluding rare words that appear 5 times max. as well as words that appear often, but do not add meaningful information, as for example the party names. This step is crucial for the later analysis: The topic model algorithm itself cannot assess any important information from word itself, but is only assessing importance based on the frequency of words. Hence, if we would not include the substantially irrelevant stopwords, our topics would most likely be composed from only stopwords.

For removing stopwords, we will use quantedas build-in tokens_remove function. I noticed, however, that some stopwords are not removed using that function. Therefore, I downloaded an additional document containing over 500 German stopwords and additionally exlcuded these from the corpus. This document can be found here: https://countwordsfree.com/stopwords/german. For reproducibility of the code, I commented out the lines that use the additional document. It only affects the later analysis to little extent.

Moreover, there are some words that are correctly not considered stopwords, but would still distort our analysis since they (a) appear very frequently across manifestos but similarly are (b) not substantially important for the latent topics. These words mostly include party abbreviations our parts of a party's name. I will remove these words manually in the next preprocessing steps.

Lastly, stemming ensures that words with a similar word stem (and therefore, with probably a similar meaning) are treated as identical instead of treating them as distinct words.

```
#additional_stopwords <- readLines("stop_words_german.txt")

preprocessing98 <- corpus1998 %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = stopwords("german")) %>%
#  tokens_remove(pattern = additional_stopwords) %>%
  tokens_wordstem(language = "de") %>%
  tokens_remove(c("bündnis", "bundnis","fdp", "f.d.p", "cdu", "csu", "spd", "pds", "grünen", "programm"
  dfm() %>%
  dfm_trim(min_termfreq = 5) %>%
  dfm_group(party)

preprocessing98
```

```
## Document-feature matrix of: 5 documents, 1,949 features (25.93% sparse) and 15 docvars.
##        features
## docs    grun wechsel bundestagswahl inhalt umwelt- naturschutz basis okolog
##   41113  229      11              2      3      15           5     6     89
##   41221    3       0              2      2       0           0     0     49
##   41320    0       5              5      0       0           0     4     25
##   41420    0       1              0      4       0          10     2     12
##   41521    2       0              1      1       1           0     1      2
##        features
## docs    wirtschaft nachhalt
##   41113         57       38
##   41221         24        6
##   41320         47       11
##   41420         55        8
##   41521         20        2
## [ reached max_nfeat ... 1,939 more features ]
```

```
preprocessing02 <- corpus2002 %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = stopwords("german")) %>%
#  tokens_remove(pattern = additional_stopwords) %>%
  tokens_wordstem(language = "de") %>%
  tokens_remove(c("bündnis", "bundnis", "fdp", "f.d.p", "cdu", "csu", "spd", "pds", "grünen", "programm
  dfm() %>%
  dfm_trim(min_termfreq = 5) %>%
  dfm_group(party)
```

```
preprocessing02
```

```
## Document-feature matrix of: 5 documents, 2,099 features (20.04% sparse) and 16 docvars.
##        features
## docs    grun partei okolog modernisier sozial wirtschaft erneuer gesellschaft
##   41113   46     19     79          15     87         40      12           61
##   41221    2     15     16           1     90         29       3           69
##   41320    1      3     11           9     53         53      10           65
##   41420    4     26     17           5     57         80       1           48
##   41521    1      2      6           1     35         62       0           40
##        features
## docs    demokratisier steh
##   41113             5   29
##   41221             1   11
##   41320             0   14
##   41420             1   20
##   41521             0    5
## [ reached max_nfeat ... 2,089 more features ]
```

As we can retrieve from the resulting document feature matrices, we see that the both corpora include five distinct documents (manifestos) and 1,853 (in 1998) vs. 2,011 (in 2002) features, which are distinct tokens/word stems.

2. Research question

After the preprocessing steps, we can start the modelling part. As already outlined, I am looking at the major German parties prior to the Federal elections 1998 and 2002, with a special emphasis on SPD and Grüne, since these parties won the Federal election 1998 and therefore introduced a change in government after 16 years of a conservative-liberal government. Using a topic modelling approach, I want to evaluate (a) What were the most prevalent topics prior to the elections 1998 and 2002? and (b) Did the SPD and Grüne have a programmatic overlap that could potentially explain their collaboration after 1998 as well as their loss of votes in 2002?

3. Topic model development

In order to answer the previously stated research question, we are using Latent Dirichlet Allocation (LDA) in order to identify broader topics within the manifestos. After excluding words (tokens) that do not add substantial meaning to the manifestos, the LDA model groups words that appear often in close proximity to each other into topics. We can chose the number of topics that the model should identify (which makes it a hyperparameter). However, including too many topics will result in a less useful output, since quite intuitively, the number of topics a manifesto can include is naturally limited. Depending on the ideological stance and programmatic focal points of a party, the LDA model will identify different topics for different parties. Intuitively, the model might identify topics related to ecological considerations in case of a Green party, and topics centered around social considerations in case of a center-left party.

As a first step, I am running the LDA model for both years and over all party manifestos and try to identify the five most important topics. As already outlined, the number of topics the model is identifying is a hyperparameter and thus can be adjusted. I tried several values for k (5, 10, 15) and noticed that adding more than five topics does not add any more meaning, since topics then become increasingly similar.

```
library(topicmodels)

lda98 <- LDA(preprocessing98, 5)
lda02 <- LDA(preprocessing02, 5)
```

This next chunk prints out the five most important topics and its corresponding tokens.

```
library(tidytext)

topic_words98 <- tidy(lda98, matrix="beta") %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

topic_words98
```

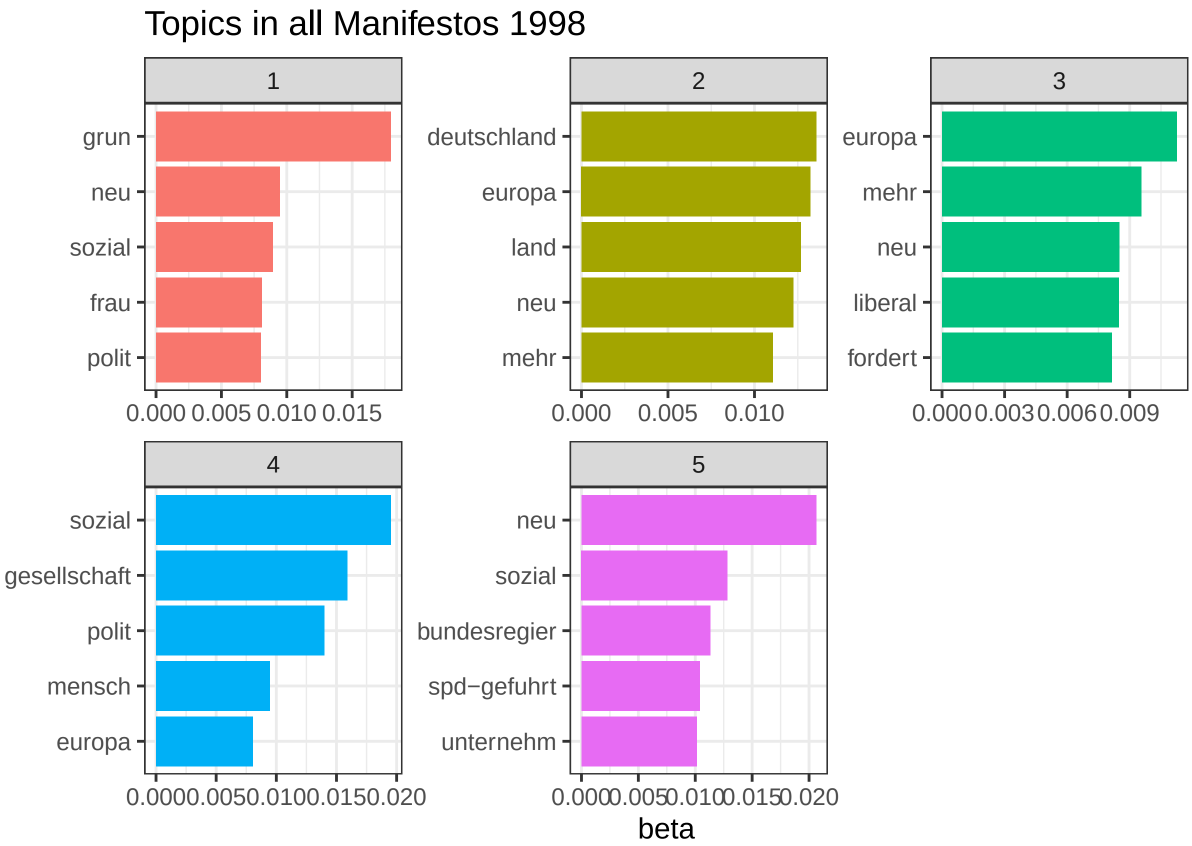
```
## # A tibble: 25 x 3
##    topic term           beta
##    <int> <chr>         <dbl>
##  1     1 grun        0.0179
##  2     1 neu         0.00948
##  3     1 sozial      0.00893
##  4     1 frau        0.00807
##  5     1 polit       0.00799
##  6     2 deutschland 0.0136
##  7     2 europa      0.0132
##  8     2 land        0.0127
##  9     2 neu         0.0122
## 10     2 mehr        0.0111
## # i 15 more rows
```

```
topic_words02 <- tidy(lda02, matrix="beta") %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

topic_words02
```

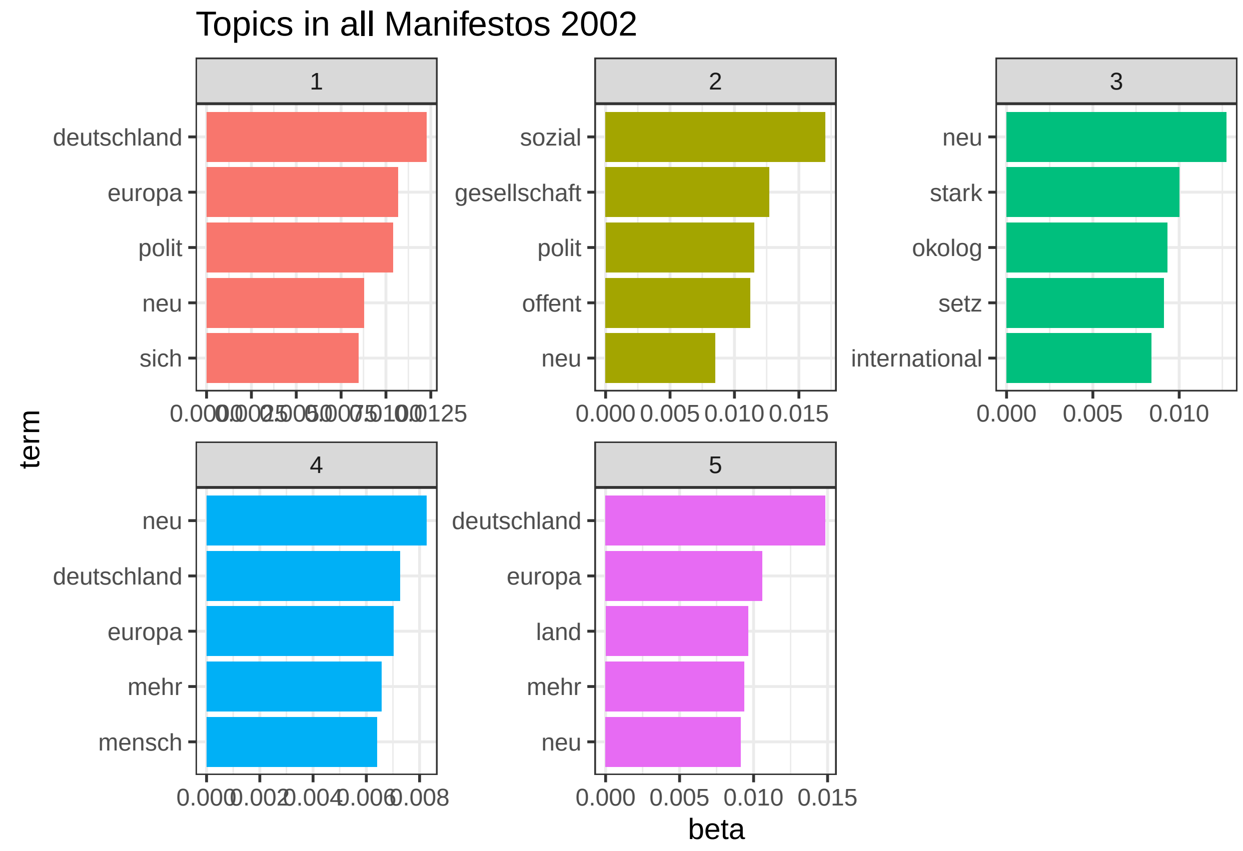
```
## # A tibble: 25 x 3
##    topic term            beta
##    <int> <chr>          <dbl>
##  1     1 deutschland  0.0123
##  2     1 europa       0.0107
##  3     1 polit        0.0104
##  4     1 neu          0.00878
##  5     1 sich         0.00846
##  6     2 sozial       0.0171
##  7     2 gesellschaft 0.0127
##  8     2 polit        0.0115
##  9     2 offent       0.0112
## 10      2 neu           0.00852
## # i 15 more rows
```

For a more intuitive visualization, the next plots shos the tokens of the mos prevalent five topics for both years and including all manifestos.

```
topic_plot98 <- topic_words98 %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  labs(title = "Topics in all Manifestos 1998") +
  ylab(label = "") +
  theme_bw()

topic_plot98
```



```
topic_plot02 <- topic_words02 %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  labs(title = "Topics in all Manifestos 2002") +
  theme_bw()

topic_plot02
```



As we can see from the output, the model successfully identified the 5 most prevalent topics across all manifestos for both election years. In 1998, most topics included some notion of Europe, which makes sense considering the transformation processes within the EU during that time. In 2002, Europe still is a prevalent token, but also new tokens as ecological and society appear. However, the topics for both years are not as distinct and clear as I expected them to be. However, this might be due to the fact that five parties with very distinct ideological preferences are included in the corpus.

For evaluating the states research question, I will now preprocess and run LDA models for the SPD and Grüne for both election years (1998 and 2002). The code applies the same preprocessing steps as before, making sure that punctuation, stopwords and less frequent or meaningful words are not influencing the analysis. Additionally, the LDA algorithm is directly applied in the next chunk as well.

```
spd98 <- corpus1998 %>%
  corpus_subset(party == 41320) %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = stopwords("german")) %>%
 # tokens_remove(pattern = additional_stopwords) %>%
  tokens_wordstem(language = "de") %>%
  tokens_remove(c("bündnis", "bundnis", "fdp", "f.d.p", "cdu", "csu", "spd", "pds", "grünen", "programm
  dfm() %>%
  dfm_trim(min_termfreq = 5) %>%
    dfm_group(party)

spd02 <- corpus2002 %>%
  corpus_subset(party == 41320) %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = stopwords("german")) %>%
 # tokens_remove(pattern = additional_stopwords) %>%
  tokens_wordstem(language = "de") %>%
  tokens_remove(c("bündnis", "bundnis","fdp", "f.d.p", "cdu", "csu", "spd", "pds", "grünen", "programm"
  dfm() %>%
  dfm_trim(min_termfreq = 5) %>%
  dfm_group(party)

lda98_spd <- LDA(spd98, 5)
lda02_spd <- LDA(spd02, 5)


topic_words98_spd <- tidy(lda98_spd, matrix="beta") %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

topic_words02_spd <- tidy(lda02_spd, matrix="beta") %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

The next chunk only stores the plots for the SPD topics in 1998 and 2002 in separate objects. The output will be discussed later and in direct comparison to the topics of Grüne.

```
spd_98_plot <- topic_words98_spd %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  labs(title = "SPD 1998") +
  ylab(label = "") +
  theme_bw()

spd_02_plot <- topic_words02_spd %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  labs(title = "SPD 2002") +
  ylab(label = "") +
  theme_bw()
```

Here, we are applying all preprocessing steps to only the manifestos of Grüne in 1998 and 2002.

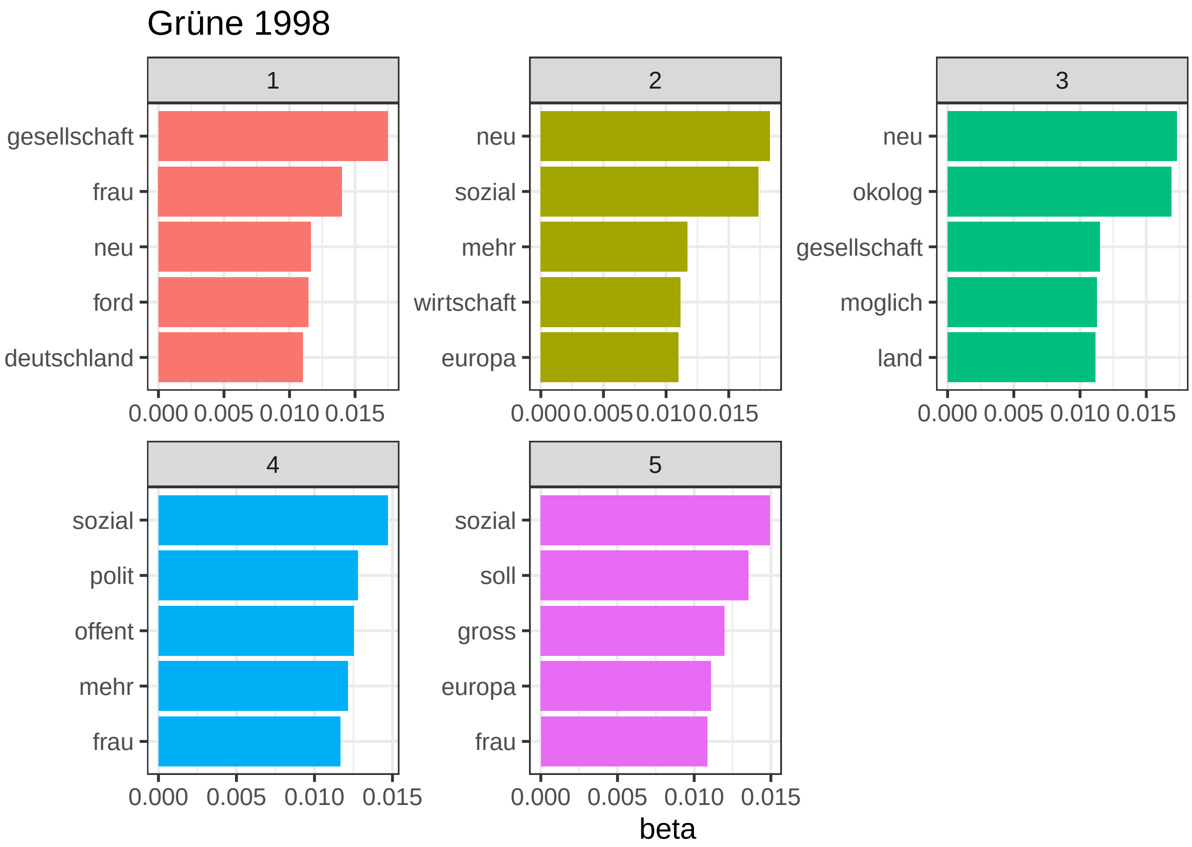
```
grüne98 <- corpus1998 %>%
  corpus_subset(party == 41113) %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = stopwords("german")) %>%
#  tokens_remove(pattern = additional_stopwords) %>%
  tokens_wordstem(language = "de") %>%
  tokens_remove(c("bündnis", "bundnis", "fdp", "f.d.p", "cdu", "csu", "spd", "pds", "grünen","grun", "p
  dfm() %>%
  dfm_trim(min_termfreq = 5) %>%
  dfm_group(party)

grüne02 <- corpus2002 %>%
  corpus_subset(party == 41113) %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = stopwords("german")) %>%
#  tokens_remove(pattern = additional_stopwords) %>%
  tokens_wordstem(language = "de") %>%
  tokens_remove(c("bündnis", "bundnis","fdp", "f.d.p", "cdu", "csu", "spd", "pds", "grünen", "grun", "p
  dfm() %>%
  dfm_trim(min_termfreq = 5) %>%
  dfm_group(party)

lda98_grüne <- LDA(grüne98, 5)
lda02_grüne <- LDA(grüne02, 5)


topic_words98_grüne <- tidy(lda98_grüne, matrix="beta") %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)

topic_words02_grüne <- tidy(lda02_grüne, matrix="beta") %>%
  group_by(topic) %>%
  slice_max(beta, n = 5) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

Again, storing the plots in two distinct objects:

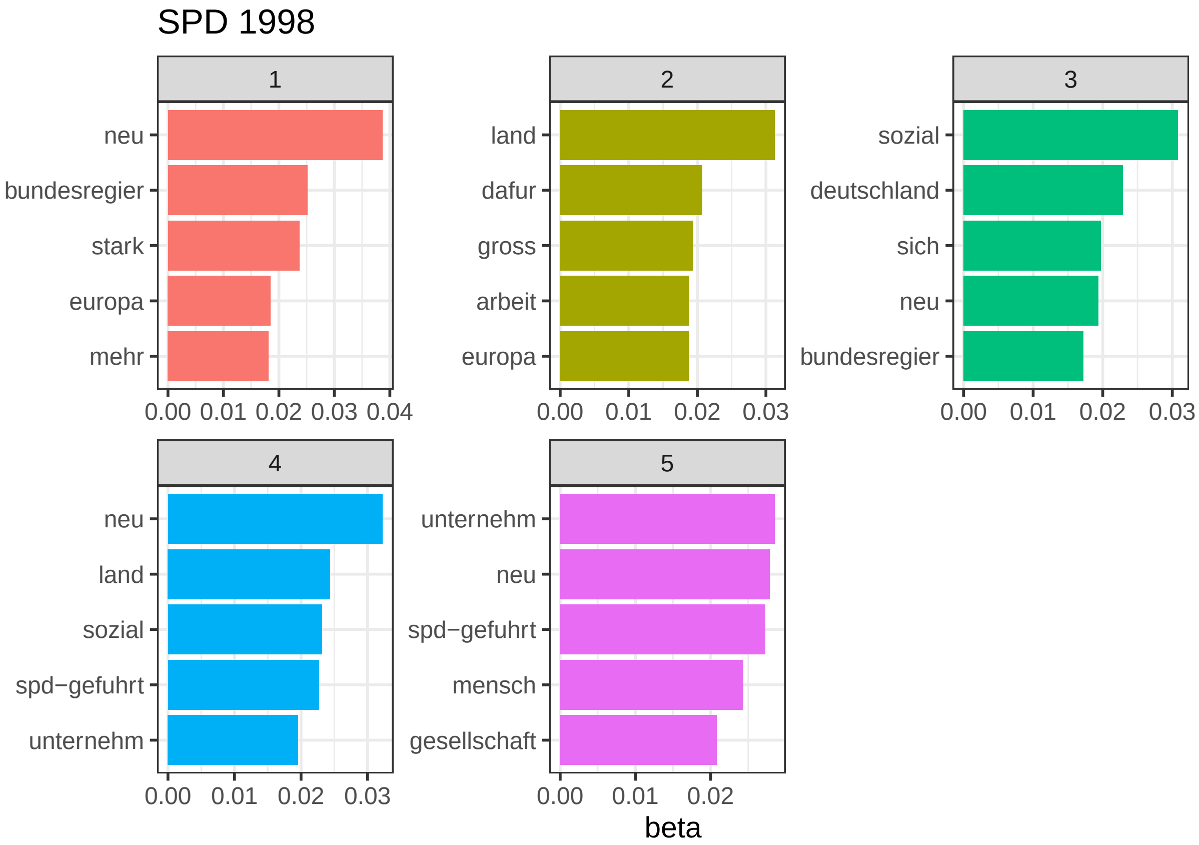
```
grüne_98_plot <- topic_words98_grüne %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  labs(title = "Grüne 1998") +
  ylab(label = "") +
  theme_bw()

grüne_02_plot <- topic_words02_grüne %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered() +
  labs(title = "Grüne 2002") +
  ylab(label = "") +
  theme_bw()
```

4. Topic Model Description

```
grüne_98_plot
```

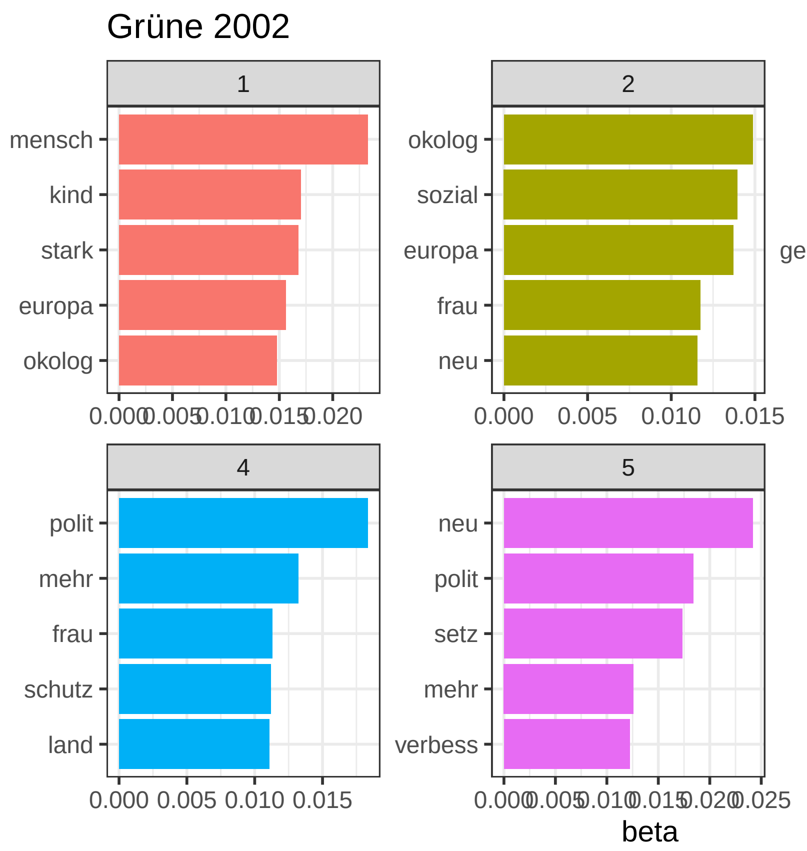


```
spd_98_plot
```

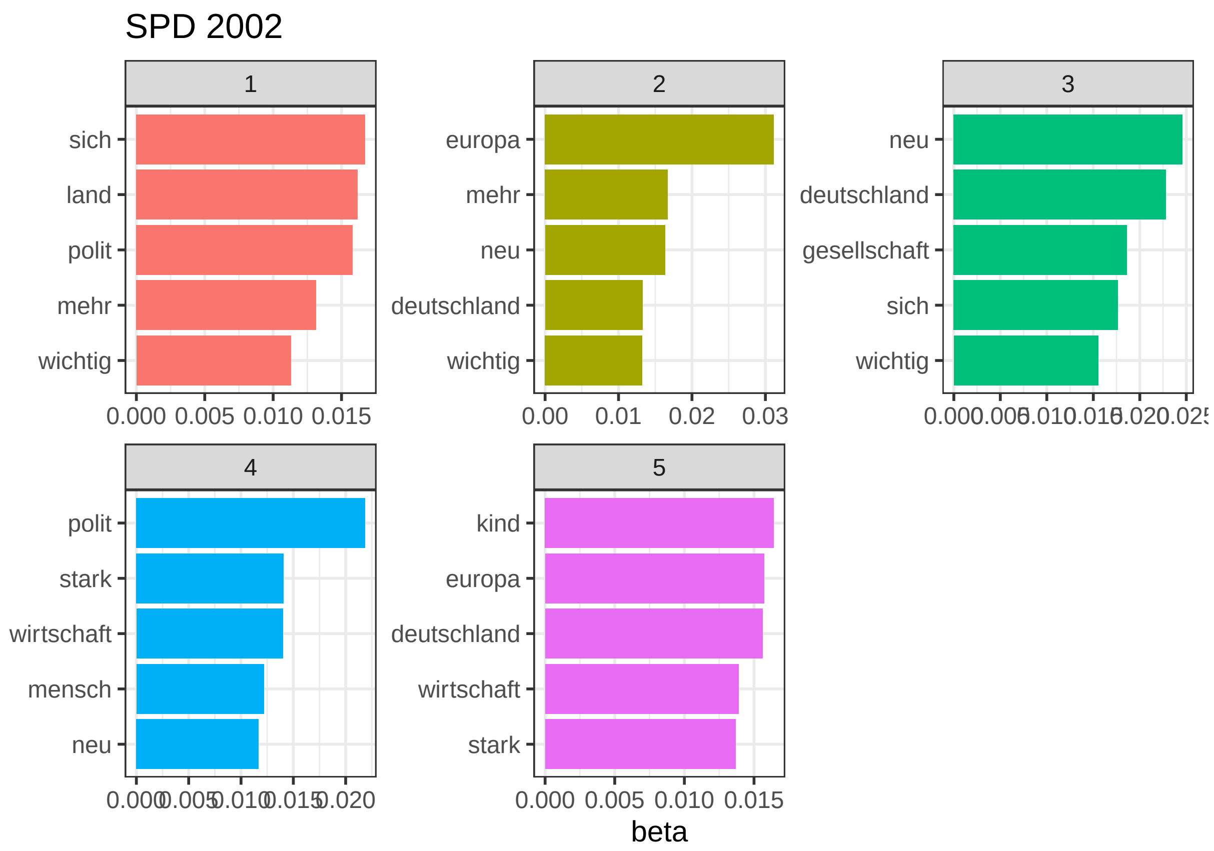


Looking at the output of the topic models that the LDA model produced for SPD and Grüne for their manifestos prior to the Federal election in 1998, we easily can see that there are some differences to the earlier model output that considered all manifestos for 1998: Exemplarily, the Green manifestos appear to contain some focus on ecological (topic 2 and 3), social (topic 2 and 5) questions as well as considerations regarding Europe (1 and 3). The focus of the SPD's manifesto is seemingly more targeted on societal/social considerations (1, 3, 4, 5), but also includes Europe (2 and 5). In summary, the topic models generates quite similar topics for both parties, but also highlights some differences (ecological considerations only for Grüne, justice only for SPD). In comparison to the full 1998 model, the results on the party level are a bit more centered around the ideological focus of both parties. On the other hand, the topics still are quite repetitive. As an intermediate conclusion and with regard to the research question, the 1998 topic model for SPD and Grüne shows that there is some programmatic overlap between the topics for both parties. However, these results should not be overestimated, since the five topics are not mutually exclusive and it is hard to assess what focus they are actually depicting. Regarding the first part of my research question, one could already conclude that we can identify some programmatic overlap which might have been important for the parties during the negotiations of an coalition agreement. Nevertheless, we must still acknowledge that a topic model only identifies prevalent topics based on frequency, and does not contain any analysis of sentiment. Moreover, it should be obvious that a collaboration on governmental level is not entirely based on programmatic overlap. Hence, a topic model can be helpful for a quick summary of programmatic focal points and the identification of ideological preferences, but only partially explain why parties choose to working together.

```
grüne_02_plot
```


Grüne 2002
1
mensch
kind
stark
europa
okolog
0.000
0.005
0.010
0.015
0.020
2
okolog
sozial
europa
frau
neu
0.000
0.005
0.010
0.015
3
neu
gerecht
gesellschaft
stark
okolog
0.000
0.005
0.010
0.015
0.020
4
polit
mehr
frau
schutz
land
0.000
0.005
0.010
0.015
5
neu
polit
setz
mehr
verbess
0.000
0.005
0.010
0.015
0.020
0.025
beta

```
spd_02_plot
```



The output of the 2002 models for Grüne and SPD shows that there might have been a slight programmatic shift to economic considerations in case of the SPD (mentions "Wirtschaft" or "Unternehmen" in topic 1 and 2, which was not present in 1998). Europe and societal considerations still seem to be important (topic 3, 4 and 5). The program of the Grüne is quite similar to the one from 1998, as environmental (topic 5), equality (2 and 4) and European (3 and 4) policies still seem to play an important role.

Summarising, the topic models for both years showed that the manifestos of SPD and Grüne indeed had some overlap, especially when it comes to the notion of Europe. As already outlined, this does not, however, include any kind of in-depth sentiment or more sophisticated content analysis. Theoretically, it could be the case that both parties mention Europe a lot, but one of them does so very positively, while the other one has negative views (although this was not the case here). When comparing the party-level topic models to the full model, it becomes evident that the topic model is able to highlight some ideological preferences, though. The model that considered the whole corpus covered a wide range of topics, while the party-level model appears somewhat more coherent.

For improving this work, one could for example include the cmp codes from the manifesto main data, which assign content specific codes to each quasi-sentence. This would allow to better evaluate the proportion of each topic in the manifestos. Moreover, applying some sentiment analysis techniques like VADER could enable us to better evaluate whether parties take positive, negative or neutral views of the identified topics.