Session 6 Topic Modelle

Christian Kahmann

12 10 2020

This exercise demonstrates the use of topic models on a text corpus for the extraction of latent semantic contexts in the documents. In this exercise we will:

1. Read in and preprocess text data,
2. Calculate a topic model using the R package *topmicmodels* and analyze its results in more detail,
3. Visualize the results from the calculated model and
4. Select documents based on their topic composition.

The process starts as usual with the reading of the corpus data. Change to your working directory, create a new R script, load the quanteda-package and define a few already known default variables.

```
# setwd("Your work directory")
options(stringsAsFactors = FALSE)
library(quanteda)
require(topicmodels)
```

The 231 SOTU addresses are rather long documents. Documents lengths clearly affects the results of topic modeling. For very short texts (e.g. Twitter posts) or very long texts (e.g. books), it can make sense to concatenate/split single documents to receive longer/shorter textual units for modeling.

For the SOTU speeches for instance, we infer the model based on paragraphs instead of entire speeches. By manual inspection / qualitative inspection of the results you can check if this procedure yields better (interpretable) topics. In `sotu_paragraphs.csv`, we provide a paragraph separated version of the speeches.

For text preprocessing, we remove stopwords, since they tend to occur as "noise" in the estimated topics of the LDA model.

```
textdata <- read.csv("data/sotu_paragraphs.csv", sep = ";", encoding = "UTF-8")

corpus <- corpus(textdata$text, docnames = textdata$doc_id)

# Build a dictionary of lemmas
lemma_data <- read.csv("resources/baseform_en.tsv", encoding = "UTF-8")

# extended stopword list
stopwords_extended <- readLines("resources/stopwords_en.txt", encoding = "UTF-8")

# Create a DTM (may take a while)
corpus_tokens <- corpus %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE, remove_symbols = TRUE) %>%
  tokens_tolower() %>%
  tokens_replace(lemma_data$inflected_form, lemma_data$lemma) %>%
  tokens_remove(pattern = stopwords_extended, padding = T)

sotu_collocations <- textstat_collocations(corpus_tokens, min_count = 25)
sotu_collocations <- sotu_collocations[1:250, ]

corpus_tokens <- tokens_compound(corpus_tokens, sotu_collocations)
```

Model calculation

After the preprocessing, we have two corpus objects: `processedCorpus`, on which we calculate an LDA topic model @blei_latent_2003. To this end, stopwords were removed, words were stemmed and converted to lowercase letters and special characters were removed. The second Corpus object `corpus` serves to be able to view the original texts and thus to facilitate a qualitative control of the topic model results.

We now calculate a topic model on the `processedCorpus`. For this purpose, a DTM of the corpus is created. In this case, we only want to consider terms that occur with a certain minimum frequency in the body. This is primarily used to speed up the model calculation.

```
# Create DTM, but remove terms which occur in less than 0.5% of all documents
DTM <- corpus_tokens %>%
  tokens_remove("") %>%
  dfm() %>%
  dfm_trim(min_docfreq = 0.005, max_docfreq = Inf, docfreq_type = "prop")

# have a look at the number of documents and terms in the matrix
dim(DTM)
```

```
## [1] 21334  1287
```

For topic modeling not only language specific stop words may beconsidered as uninformative, but also domain specific terms. We remove 10 of the most frequent terms to improve the modeling.

```
top10_terms <- c( "unite_state", "past_year", "year_ago", "year_end", "government", "state", "country",

DTM <- DTM[, !(colnames(DTM) %in% top10_terms)]

# due to vocabulary pruning, we have empty rows in our DTM
# LDA does not like this. So we remove those docs from the
# DTM and the metadata
sel_idx <- rowSums(DTM) > 0
DTM <- DTM[sel_idx, ]
textdata <- textdata[sel_idx, ]
```

As an unsupervised machine learning method, topic models are suitable for the exploration of data. The calculation of topic models aims to determine the proportionate composition of a fixed number of topics in the documents of a collection. It is useful to experiment with different parameters in order to find the most suitable parameters for your own analysis needs.

For parameterized models such as Latent Dirichlet Allocation (LDA), the number of topics `K` is the most important parameter to define in advance. How an optimal `K` should be selected depends on various factors. If `K` is too small, the collection is divided into a few very general semantic contexts. If `K` is too large, the collection is divided into too many topics of which some may overlap and others are hardly interpretable.

For our first analysis we choose a thematic "resolution" of `K = 20` topics. In contrast to a resolution of 100 or more, this number of topics can be evaluated qualitatively very easy.

```
# load package topicmodels
require(topicmodels)
# number of topics
K <- 20
# set random number generator seed
set.seed(9161)
# compute the LDA model, inference via 1000 iterations of Gibbs sampling
topicModel <- LDA(DTM, K, method="Gibbs", control=list(iter = 500, verbose = 25))
```

Depending on the size of the vocabulary, the collection size and the number K, the inference of topic models can take a very long time. This calculation may take several minutes. If it takes too long, reduce the vocabulary in the DTM by increasing the minimum frequency in the previous step.

The topic model inference results in two (approximate) posterior probability distributions: a distribution `theta` over K topics within each document and a distribution `beta` over V terms within each topic, where V represents the length of the vocabulary of the collection (V = 1278). Let's take a closer look at these results:

```
# have a look a some of the results (posterior distributions)
tmResult <- posterior(topicModel)
# format of the resulting object
attributes(tmResult)
```

```
## $names
## [1] "terms"  "topics"
```

```
ncol(DTM)                # lengthOfVocab
```

```
## [1] 1278
```

```
# topics are probability distribtions over the entire vocabulary
beta <- tmResult$terms   # get beta from results
dim(beta)                # K distributions over ncol(DTM) terms
```

```
## [1]   20 1278
```

```
rowSums(beta)            # rows in beta sum to 1
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
nrow(DTM)               # size of collection
```

```
## [1] 20950
```

```
# for every document we have a probability distribution of its contained topics
theta <- tmResult$topics
dim(theta)               # nDocs(DTM) distributions over K topics
```

```
## [1] 20950    20
```

```
rowSums(theta)[1:10]     # rows in theta sum to 1
```

```
##  1  2  3  4  5  6  7  8  9 10
##  1  1  1  1  1  1  1  1  1  1
```

Let's take a look at the 10 most likely terms within the term probabilities `beta` of the inferred topics (only the first 8 are shown below).

```
terms(topicModel, 10)
```

```
##       Topic 1     Topic 2      Topic 3      Topic 4     Topic 5
##  [1,] "time"      "law"        "interest"   "people"    "economy"
##  [2,] "great"     "citizen"    "trade"      "nation"    "child"
##  [3,] "result"    "case"       "american"   "principle" "american"
##  [4,] "condition" "justice"    "commerce"   "spirit"    "job"
##  [5,] "indian"    "property"   "relation"   "political" "family"
##  [6,] "change"    "court"      "foreign"    "great"     "million"
##  [7,] "progress"  "person"     "commercial" "interest"  "education"
##  [8,] "continue"  "protection" "policy"     "power"     "budget"
##  [9,] "present"   "provision"  "open"       "national"  "school"
```

```
## [10,] "general"   "question"   "europe"     "feel"      "reform"
##       Topic 6          Topic 7      Topic 8
##  [1,] "national"       "great"      "public"
##  [2,] "program"        "good"       "bank"
##  [3,] "federal"        "business"   "issue"
##  [4,] "system"         "public"     "money"
##  [5,] "administration" "control"    "system"
##  [6,] "provide"        "man"        "time"
##  [7,] "development"    "condition"  "gold"
##  [8,] "resource"       "action"     "credit"
##  [9,] "improve"        "matt"       "return"
## [10,] "policy"         "individual" "currency"
```

For the next steps, we want to give the topics more descriptive names than just numbers. Therefore, we simply concatenate the five most likely terms of each topic to a string that represents a pseudo-name for each topic.

```
top5termsPerTopic <- terms(topicModel, 5)
topicNames <- apply(top5termsPerTopic, 2, paste, collapse=" ")
```

Visualization of Words and Topics

Although wordclouds may not be optimal for scientific purposes they can provide a quick visual overview of a set of terms. Let's look at some topics as wordcloud.

In the following code, you can change the variable **topicToViz** with values between 1 and 20 to display other topics.

```
require(wordcloud2)
# visualize topics as word cloud
topicToViz <- 11 # change for your own topic of interest
topicToViz <- grep('mexico', topicNames)[1] # Or select a topic by a term contained in its name
# select to 40 most probable terms from the topic by sorting the term-topic-probability vector in decre
top40terms <- sort(tmResult$terms[topicToViz,], decreasing=TRUE)[1:40]
words <- names(top40terms)
# extract the probabilites of each of the 40 terms
probabilities <- sort(tmResult$terms[topicToViz,], decreasing=TRUE)[1:40]
# visualize the terms as wordcloud
wordcloud2(data.frame(words, probabilities), shuffle = FALSE, size = 0.8)
```

Let us now look more closely at the distribution of topics within individual documents. To this end, we visualize the distribution in 3 sample documents.

Let us first take a look at the contents of three sample documents:

```
exampleIds <- c(2, 100, 200)
cat(corpus[exampleIds[1]])
cat(corpus[exampleIds[2]])
cat(corpus[exampleIds[3]])
```

```
## 2: I embrace with great satisfaction the opportunity which now presents itself
## of congratulating you on the present favorable prospects of our public
## affairs. The recent accession of the important state of North Carolina to
## the Constitution of the United States (of which official information has
## been received), the rising credit and respectability of our country, the
## general and increasing good will ...
```

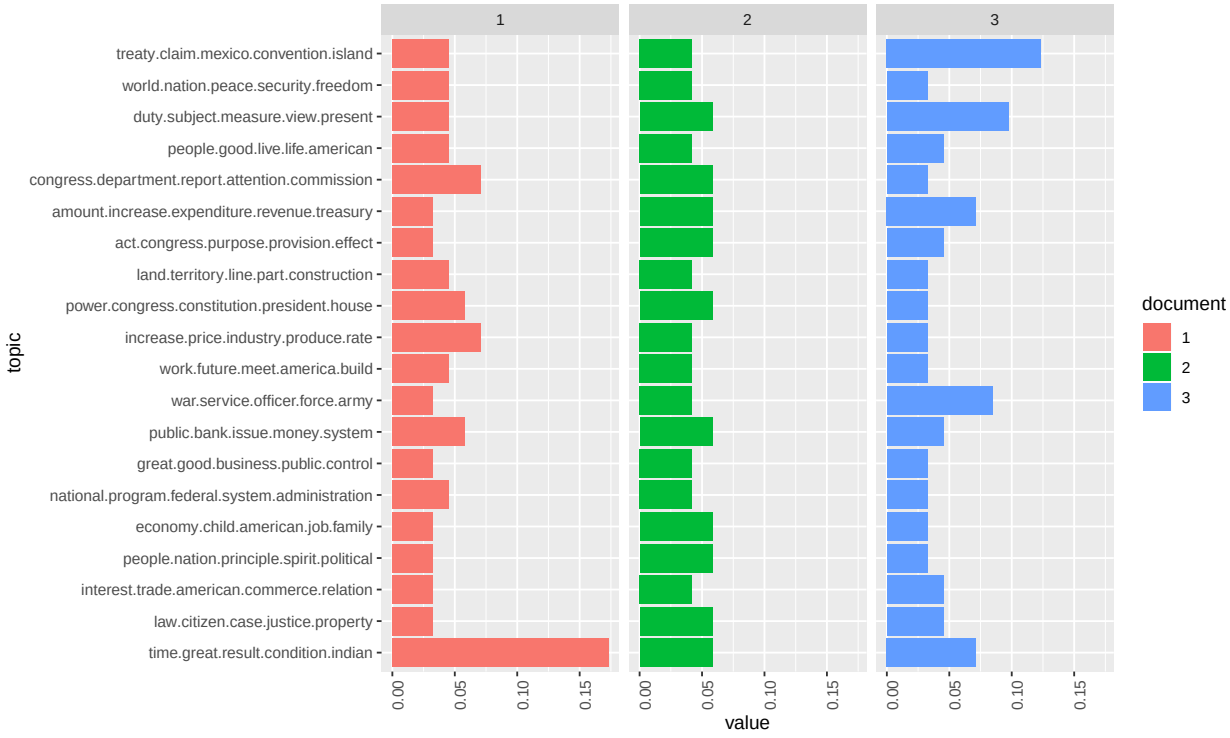
```
## 100: Provision is likewise requisite for the reimbursement of the loan which has
## been made of the Bank of the United States, pursuant to the eleventh
## section of the act by which it is incorporated. In fulfilling the public
## stipulations in this particular it is expected a valuable saving will be
## made....
```

```
## 200: After many delays and disappointments arising out of the European war, the
## final arrangements for fulfilling the engagements made to the Dey and
## Regency of Algiers will in all present appearance be crowned with success,
## but under great, though inevitable, disadvantages in the pecuniary
## transactions occasioned by that war, which will render further provision
## necessary. The actual liberation of all ...
```

After looking into the documents, we visualize the topic distributions within the documents.

```r
# load libraries for visualization
library("reshape2")
library("ggplot2")
N <- length(exampleIds)
# get topic proportions form example documents
topicProportionExamples <- theta[exampleIds,]
colnames(topicProportionExamples) <- topicNames
vizDataFrame <- melt(cbind(data.frame(topicProportionExamples), document = factor(1:N)), variable.name =

ggplot(data = vizDataFrame, aes(topic, value, fill = document), ylab = "proportion") +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip() +
  facet_wrap(~ document, ncol = N)
```



Topic distributions

The figure above shows how topics within a document are distributed according to the model. In the current model all three documents show at least a small percentage of each topic. However, two to three topics dominate each document.

The topic distribution within a document can be controlled with the *Alpha*-parameter of the model. Higher alpha priors for topics result in an even distribution of topics within a document. Low alpha priors ensure that the inference process distributes the probability mass on a few topics for each document.

In the previous model calculation the alpha-prior was automatically estimated in order to fit to the data (highest overall probability of the model). However, this automatic estimate does not necessarily correspond

to the results that one would like to have as an analyst. Depending on our analysis interest, we might be interested in a more peaky/more even distribution of topics in the model.

Now let us change the alpha prior to a lower value to see how this affects the topic distributions in the model.

```
# see alpha from previous model
attr(topicModel, "alpha")
```

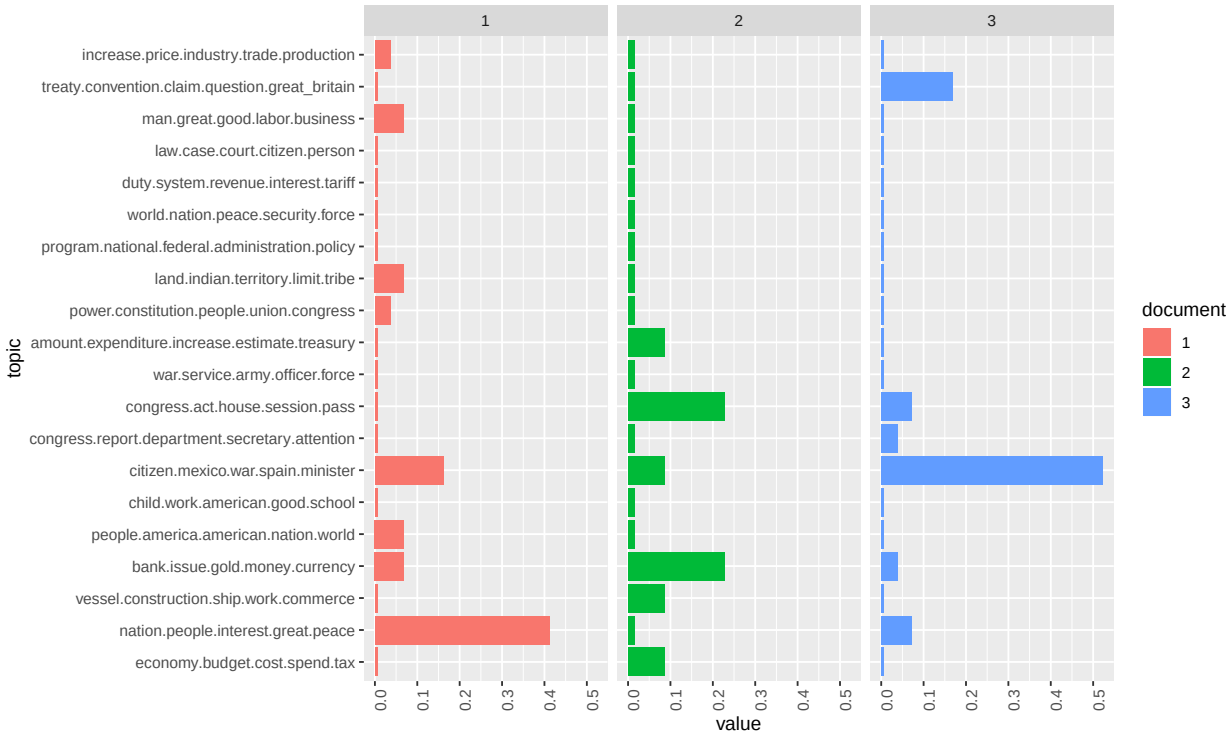
```
## [1] 2.5
```

```
topicModel2 <- LDA(DTM, K, method="Gibbs", control=list(iter = 500, verbose = 25, alpha = 0.2))
tmResult <- posterior(topicModel2)
theta <- tmResult$topics
beta <- tmResult$terms
topicNames <- apply(terms(topicModel2, 5), 2, paste, collapse = " ")  # reset topicnames
```

Now visualize the topic distributions in the three documents again. What are the differences in the distribution structure?

```
# get topic proportions form example documents
topicProportionExamples <- theta[exampleIds,]
colnames(topicProportionExamples) <- topicNames
vizDataFrame <- melt(cbind(data.frame(topicProportionExamples), document = factor(1:N)), variable.name =

ggplot(data = vizDataFrame, aes(topic, value, fill = document), ylab = "proportion") +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip() +
  facet_wrap(~ document, ncol = N)
```



Topic ranking

First, we try to get a more meaningful order of top terms per topic by re-ranking them with a specific score @Chang.2012. The idea of re-ranking terms is similar to the idea of TF-IDF. The more a term appears in top levels w.r.t. its probability, the less meaningful it is to describe the topic. Hence, the scoring favors less general, more specific terms to describe a topic.

```
# re-rank top topic terms for topic names
topicNames <- apply(lda::top.topic.words(beta, 5, by.score = T), 2, paste, collapse = " ")
```

What are the defining topics within a collection? There are different approaches to find out which can be used to bring the topics into a certain order.

Approach 1: We sort topics according to their probability within the entire collection:

```
# What are the most probable topics in the entire collection?
topicProportions <- colSums(theta) / nrow(DTM)  # mean probablities over all paragraphs
names(topicProportions) <- topicNames     # assign the topic names we created before
sort(topicProportions, decreasing = TRUE) # show summed proportions in decreased order
```

```
##  [1] "0.06552 : america world american people nation"
##  [2] "0.06123 : world nation peace security freedom"
##  [3] "0.05601 : mexico spain minister island war"
##  [4] "0.05514 : child school american work family"
##  [5] "0.05513 : nation people peace interest great"
##  [6] "0.05445 : program federal administration national development"
##  [7] "0.05398 : report department congress secretary recommendation"
##  [8] "0.05301 : budget spend economy tax cost"
##  [9] "0.05165 : treaty convention claim great_britain negotiation"
## [10] "0.04915 : amount expenditure estimate treasury increase"
## [11] "0.04811 : constitution power union people congress"
## [12] "0.04734 : congress act house session representative"
## [13] "0.04579 : war army service officer navy"
## [14] "0.04521 : man labor business corporation public"
## [15] "0.04509 : price production industry market trade"
## [16] "0.04501 : duty tariff revenue system article"
## [17] "0.04477 : law court case person justice"
## [18] "0.04319 : vessel construction ship port commerce"
## [19] "0.04058 : indian land territory tribe public_land"
## [20] "0.03963 : bank gold currency issue money"
```

We recognize some topics that are way more likely to occur in the corpus than others. These describe rather general thematic coherences. Other topics correspond more to specific contents.

Approach 2: We count how often a topic appears as a primary topic within a paragraph This method is also called Rank-1.

```
countsOfPrimaryTopics <- rep(0, K)
names(countsOfPrimaryTopics) <- topicNames
for (i in 1:nrow(DTM)) {
  topicsPerDoc <- theta[i, ] # select topic distribution for document i
  # get first element position from ordered list
  primaryTopic <- order(topicsPerDoc, decreasing = TRUE)[1]
  countsOfPrimaryTopics[primaryTopic] <- countsOfPrimaryTopics[primaryTopic] + 1
}
sort(countsOfPrimaryTopics, decreasing = TRUE)
```

```
##  [1] "1826 : america world american people nation"
##  [2] "1524 : world nation peace security freedom"
##  [3] "1386 : budget spend economy tax cost"
##  [4] "1347 : child school american work family"
##  [5] "1317 : nation people peace interest great"
##  [6] "1315 : mexico spain minister island war"
##  [7] "1242 : report department congress secretary recommendation"
##  [8] "1209 : program federal administration national development"
##  [9] "1048 : treaty convention claim great_britain negotiation"
## [10] "995 : amount expenditure estimate treasury increase"
## [11] "915 : congress act house session representative"
## [12] "870 : constitution power union people congress"
## [13] "868 : war army service officer navy"
## [14] "857 : vessel construction ship port commerce"
## [15] "753 : bank gold currency issue money"
## [16] "737 : law court case person justice"
## [17] "726 : price production industry market trade"
## [18] "703 : duty tariff revenue system article"
## [19] "674 : man labor business corporation public"
## [20] "638 : indian land territory tribe public_land"
```

We see that sorting topics by the Rank-1 method places topics with rather specific thematic coherences in upper ranks of the list.

This sorting of topics can be used for further analysis steps such as the semantic interpretation of topics found in the collection, the analysis of time series of the most important topics or the filtering of the original collection based on specific sub-topics.

Filtering documents

The fact that a topic model conveys of topic probabilities for each document, resp. paragraph in our case, makes it possible to use it for thematic filtering of a collection. As filter we select only those documents which exceed a certain threshold of their probability value for certain topics (for example, each document which contains topic `X` to more than `Y` percent).

In the following, we will select documents based on their topic content and display the resulting document quantity over time.

```
topicToFilter <- 6  # you can set this manually ...
# ... or have it selected by a term in the topic name
topicToFilter <- grep('mexico ', topicNames)[1]
topicThreshold <- 0.1 # minimum share of content must be attributed to the selected topic
selectedDocumentIndexes <- (theta[, topicToFilter] >= topicThreshold)
filteredCorpus <- corpus %>% corpus_subset(subset = selectedDocumentIndexes)

# show length of filtered corpus
filteredCorpus
```

```
## Corpus consisting of 3,106 documents and 0 docvars.
```

Our filtered corpus contains 3106 documents related to the topic 7 to at least 10 %.

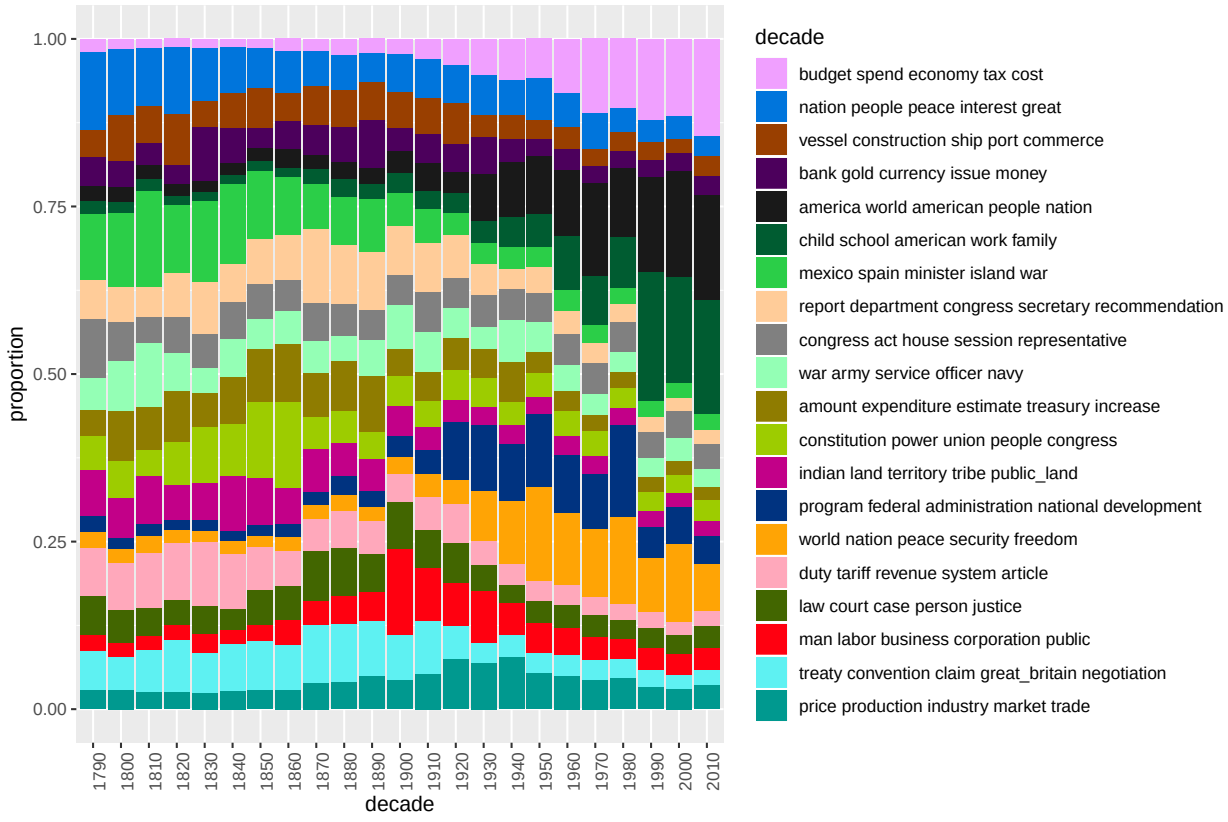
Topic proportions over time

In a last step, we provide a distant view on the topics in the data over time. For this, we aggregate mean topic proportions per decade of all SOTU speeches. These aggregated topic proportions can then be visualized, e.g. as a bar plot.

```
# append decade information for aggregation
textdata$decade <- paste0(substr(textdata$date, 0, 3), "0")
# get mean topic proportions per decade
topic_proportion_per_decade <- aggregate(theta, by = list(decade = textdata$decade), mean)
# set topic names to aggregated columns
colnames(topic_proportion_per_decade)[2:(K+1)] <- topicNames

# reshape data frame
vizDataFrame <- melt(topic_proportion_per_decade, id.vars = "decade")

# plot topic proportions per deacde as bar plot
require(pals)
ggplot(vizDataFrame, aes(x=decade, y=value, fill=variable)) +
  geom_bar(stat = "identity") + ylab("proportion") +
  scale_fill_manual(values = paste0(alphabet(20), "FF"), name = "decade") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



The visualization shows that topics around the relation between the federal government and the states as well as inner conflicts clearly dominate the first decades. Security issues and the economy are the most important topics of recent SOTU addresses.