UNIVERSITÄT LEIPZIG

Computational Humanities

UNIVERSITÄT LEIPZIG

Dr. Ing. Andreas Niekler

Exercise: Topic Models Introduction

Table of contents

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")
sotu_corpus <- corpus(textdata$text, docnames = textdata$doc_id)
# Build a dictionary of lemmas
lemma_data <- read.csv("../data/resources/baseform_en.tsv", encoding = "UTF-8")
# extended stopword list
stopwords_extended <- readLines("../data/resources/stopwords_en.txt",
    encoding = "UTF-8")
# Create a DTM (may take a while)

corpus_tokens <- sotu_corpus %>%
    tokens(remove_punct = TRUE, remove_numbers = TRUE, remove_symbols = TRUE) %>%
    tokens_tolower() %>%
    tokens_replace(lemma_data$inflected_form, lemma_data$lemma,
        valuetype = "fixed") %>%
    tokens_remove(pattern = stopwords_extended, padding = T)

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

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

1 Model calculation

After the preprocessing, we have two corpus objects: `processedCorpus`, on which we calculate an LDA topic model Blei, Ng, and Jordan [1]. 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 1%
# of all documents
DTM <- corpus_tokens %>%
    tokens_remove("") %>%
    dfm() %>%
    dfm_trim(min_docfreq = 3)
# have a look at the number of documents and terms in the
# matrix
```

```
dim(DTM)
```

```
[1] 21334  9687
```

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", "year", "make", "seek")

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. We also set the seed for the random number generator to ensure reproducible results between repeated model inferences.

```
# load package topicmodels
require(topicmodels)
# number of topics
K <- 20
# compute the LDA model, inference via n iterations of Gibbs sampling
topicModel <- LDA(DTM, K, method="Gibbs", control=list(
  iter = 500,
  seed = 1,
  verbose = 25,
  alpha = 0.02))
```

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 = 9677). 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] 9677
```

```
# 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 9677
```

```
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] 21272
```

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

```
[1] 21272    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,] "america"  "interest" "program"     "duty"        "navy"
 [2,] "great"    "nation"   "federal"     "price"       "service"
 [3,] "people"   "great"    "national"    "increase"    "army"
 [4,] "nation"   "people"   "congress"    "tariff"      "war"
 [5,] "american" "public"   "energy"      "export"      "officer"
 [6,] "world"    "power"    "health"      "manufacture" "ship"
```

```
 [7,] "good"     "peace"    "development" "market"      "man"
 [8,] "time"     "policy"   "provide"     "product"     "vessel"
 [9,] "tonight"  "good"     "work"        "trade"       "naval"
[10,] "live"     "time"     "education"   "production"  "force"
      Topic 6          Topic 7    Topic 8
 [1,] "world"          "work"     "increase"
 [2,] "nation"         "child"    "expenditure"
 [3,] "peace"          "american" "estimate"
 [4,] "economic"       "job"      "amount"
 [5,] "security"       "good"     "fiscal_year"
 [6,] "international"  "family"   "department"
 [7,] "continue"       "school"   "receipt"
 [8,] "defense"        "people"   "service"
 [9,] "trade"          "million"  "revenue"
[10,] "effort"         "america"  "mail"
```

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 = " ")
```

2 Visualization

We also can use the LDAvis package by Sievert and Shirley [3] to visualize the model you computed. You can observe the terms in the topics and see whether there are related or interesting topics in your corpus. LDAvis also calculates the importance of a topic. This is done by determine how much probability is assigned to the topics in the documents. We can sum this up and display the importance.

```
# LDAvis browser
library(LDAvis)
library("tsne")
svd_tsne <- function(x) tsne(svd(x)$u)
json <- createJSON(phi = beta, theta = theta, doc.length = rowSums(DTM),
    vocab = colnames(DTM), term.frequency = colSums(DTM), mds.method = svd_tsne,
    plot.opts = list(xlab = "", ylab = ""))
serVis(json)
```

3 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

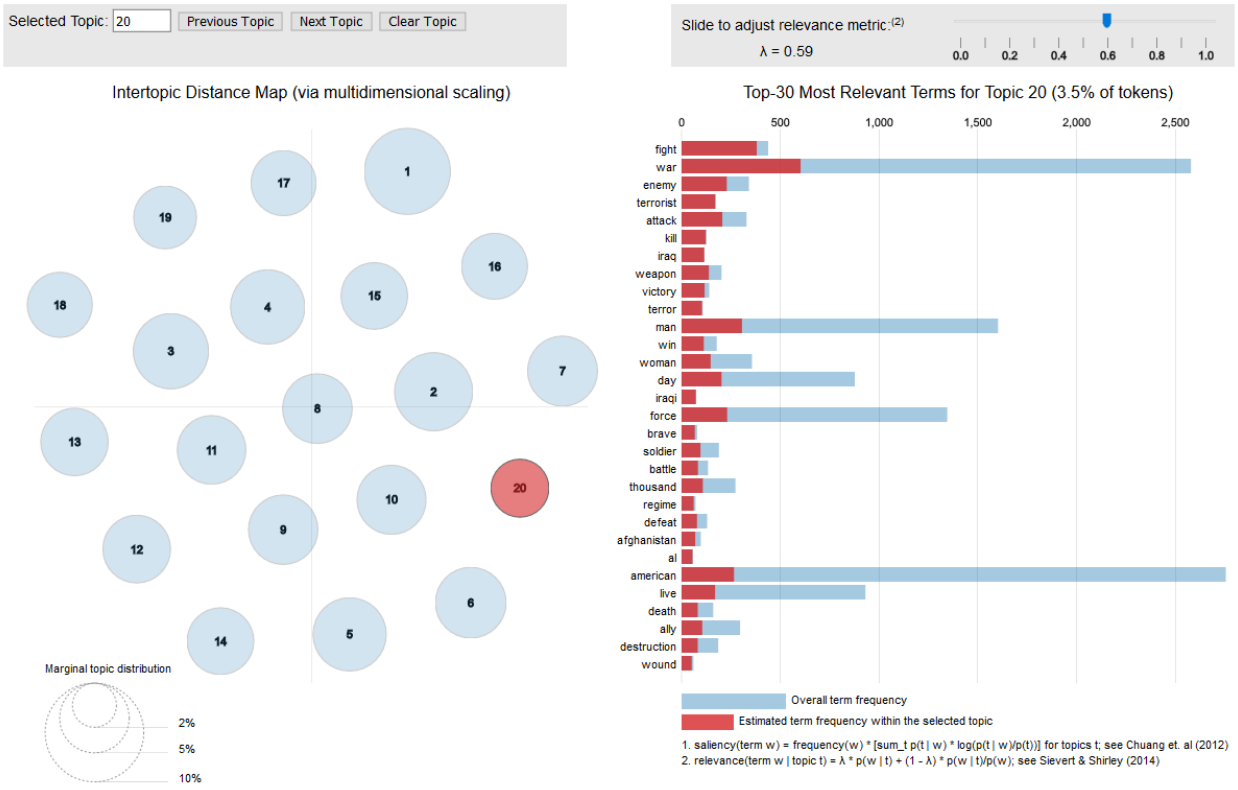


Figure 1: LDAvis example

document quantity over time.

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

```
Corpus consisting of 1,726 documents.
7 :
"There was reason to hope that the pacific measures adopted w..."

9 :
"Various considerations also render it expedient that the ter..."

27 :
```

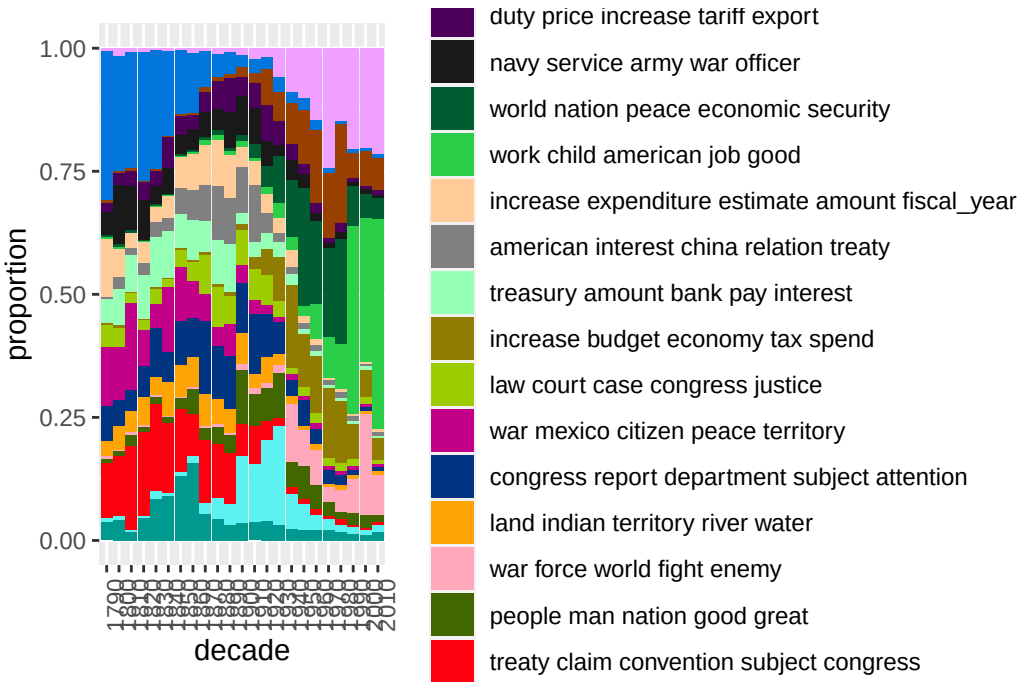
```
"It has been heretofore known to Congress that frequent incur..."

28 :
"These aggravated provocations rendered it essential to the s..."

30 :
"Your attention seems to be not less due to that particular b..."

40 :
""In vain may we expect peace with the Indians on our frontie..."

[ reached max_ndoc ... 1,720 more documents ]
```

Our filtered corpus contains 1726 documents related to the topic 13 to at least 10 %.

4 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.

```r
# 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.

For more details about topic modeling and some best practice advise, see also Maier et al. [2].

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
# Or select a topic by a term contained in its name
topicToViz <- grep("mexico", topicNames)[1]
# select to 40 most probable terms from the topic by
# sorting the term-topic-probability vector in decreasing
# order
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)
```


hostility
mexican
officer
protection
frontier
authority
property
nation
arm
act
territory
indian
receive
troop
force
citizen
law
american
give
order
texas
cuba
war
mexico
part
person
general
peace
measure
spain
army
island
power
prevent
military
vessel
command
time
continue
protect

References

[1] D.M. Blei, A.Y. Ng, and M.I. Jordan. "Latent dirichlet allocation". In: The Journal of Machine Learning Research 3 (2003), pp. 993–1022.

[2] Daniel Maier et al. "Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology". In: Communication Methods and Measures 12.2-3 (2018), pp. 93–118. DOI: 10.1080/19312458.2018.1430754.

[3] Carson Sievert and Kenneth E. Shirley. "LDAvis: A method for visualizing and interpreting topics". In: Proceedings of the workshop on interactive language learning, visualization, and interfaces. 2014, pp. 63–70. URL: http://www.aclweb.org/website/old_anthology/W/W14/W14-31.pdf#page=73 (visited on 08/31/2016).