Mini-Workshop (Structural) Topic Models

Marko Bachl

Sommersemester 2020 | IJK Hannover

Contents

Chapter 1

Überblick

1.1 Inhalt des virtuellen Mini-Workshops

- In diesem Mini-Workshop erläutere ich das praktische Vorgehen einer Datenanalyse mit *Structural Topic Models*. Wir behandeln die folgenden Schritte im Analyseprozess:
  - Modellspezifikation
  - Modellvergleich zur Auswahl eines geeigneten Modells
  - Interpretation der Topics im finalen Modell
  - Darstellung der Ergebnisse
  - Weitere Analysen
    - Identifikation verwandter Themen
    - Zusammenhänge der Themenprävalenz mit Kovariaten.
- Wir verwenden das Paket `{stm}` (Roberts et al., 2019) zum Schätzen von Topic Models. Für die Variante der *Structural* Topic Models und die Implementation in diesem Paket sprechen *für mich* die folgenden Gründe
  - Gute Integration mit *R* und Paketen, die ich für die Arbeit mit Text-Daten verwende (insbesondere `{quanteda}` und `{tidytext}`)
  - Gute ergänzende Pakete zur Arbeit mit den Modellen (insbesondere `{stminsights}`)
  - Vergleichsweise schnelle Modellschätzung auch mit großen Datensätzen
  - Direktes Schätzen von Zusammenhängen von Topics mit Kovariaten
  - Initialisieren der Modellschätzung mit dem Spectral Algorithmus
  - Recht weit verbreitet in einem Feld, in dem ich viel lese (Politische Kommunkation nach einem weitem Verständnis)
- Die Darstellung basiert auf einer Analyse, die ich gemeinsam mit Elena Link durchgeführt habe. Wir untersuchten, wie das Thema Impfen in Online-Foren für Eltern diskutiert wurde. Wir verwenden aber nur einen *nicht repräsentativen* Ausschnitt aus dem Material, um die notwendige

Rechenleistung und -zeit zu verringern.
  - Einen Preprint zur Analyse könnt ihr hier lesen: Vaccine-related Discussions in Online Communities for Parents. A Quantitative Overview.
  - Die Dokumentation zur Studie ist hier verfügbar: https://bachl.github.io/vaccine_discussions/. Daten und Analyse-Skripts gibt es im OSF. In diesem Material werde auch die Datenerhebung mittels Web-Scraping und die Datenaufbereitung erläutert. Diese Inhalte sind *nicht* Teil dieses Workshops. Wenn ihr Fragen dazu habt, dürft ihr sie natürlich stellen.

1.2 Welche Inhalte wir *nicht* behandeln

- Auch wenn das im direkten Vergleich mit dem Parallel-Angebot zu Panel Data Analysis (meine Ausführlichkeit dort sind ein Grund für die spätere Lieferung dieser Materialien) enttäuschend sein mag: Die Inhalte in diesem Mini-Workshop entsprechen in ihrem Umfang wirklich nur dem, was ich zu Beginn des Digital-Semesters geplant und angekündigt hatte. Der Mini-Workshop ersetzt keine tiefer gehende Einarbeitung in die Methode, sondern ist als ein Einstieg zu verstehen.

- Wir behandeln hier keine theoretischen, statistischen oder auf die Software-Implementierung der Modellschätzung bezogenen Fragen. Die Grundlagen dazu können aus den Texten im LMS entnommen werden (Maier et al., 2018; Roberts et al., 2019).

- Es gibt neben `{stm}` viele andere Implementationen in *R* und ihn anderer Software. Gefühlt gibt es alle 6 Monate eine neue Variante von Topic Models, alle 3 Monate eine neue Implementierung und jeden Monat ein Paket mit zusätzlichen Tools für die Arbeit mit Topic Models. Meine Entscheidung für `{stm}` ist keine informierte Entscheidung gegen andere Varianten, Implementierungen und Tools. Dieser Workshop ist keine Aufforderung, ausschließlich `{stm}` zu nutzen. Informiert euch gegebenenfalls selbst über Software-Lösungen, die für eure Bedürfnisse geeignet sind.

- Dieser Mini-Workshop ist kein *R*-Tutorial. Wenn ihr Interesse habt, *R*-Kenntnisse zu erwerben und zu vertiefen, empfehle ich R4DS.

- Dieser Mini-Workshop ist keine allgemeine Einführung in die computergestützte Inhaltsanalyse. Wenn ihr allgemein mit *R* arbeiten möchtet, empfehle ich zu diesem Thema die Einführung von Cornelius Puschmann.

1.3 Aufbau des Workshops

- Inhaltlicher Aufbau: Siehe Kapitel-Gliederung

Material

- Dieses Dokument + R Skripte: (Hoffentlich) mehr oder weniger selbsterklärendes Material
  - Kuratierte Form ist dieses HTML-Dokument
  - Es gibt auch ein PDF, das ich aber nicht formatiert habe
- Daten: Ein Ausschnitt auf den Daten der oben genannten Beispielstudie. Eine genauere Beschreibung folgt im nächsten Abschnitt.
- Screencast: Zu einigen Analyseschritten stelle ich Screencasts zur Verfügung. Diese sind größtenteils ergänzend gedacht. Bis auf wenige Ausnahmen sollte das schriftliche Material selbsterklärend sein.
- Übungen: Zu einigen Analysen gibt es Übungsaufgaben.
  - XXX

Pakete

Wir verwenden die folgenden Pakete

```
if (!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse, stm, stminsights, tidytext, quanteda, lubridate, knitr,
    tictoc, furrr)
theme_set(theme_bw())  # ggplot theme

tibble(package = c("R", sort(pacman::p_loaded()))) %>% mutate(version = map_chr(package,
    ~as.character(pacman::p_version(package = .x)))) %>% knitr::kable()
```

package	version
R	3.6.2
dplyr	0.8.4
forcats	0.4.0
furrr	0.1.0
future	1.16.0
ggplot2	3.3.1
knitr	1.28
lubridate	1.7.4
pacman	0.5.1
purrr	0.3.3
quanteda	2.0.0
readr	1.3.1
stm	1.3.5
stminsights	0.4.0
stringr	1.4.0
tibble	2.1.3
tictoc	1.0
tidyr	1.0.2
tidytext	0.2.3
tidyverse	1.3.0

Chapter 2

Beispiel-Daten und Aufbereitung

2.1 Laden der Daten und Übersicht

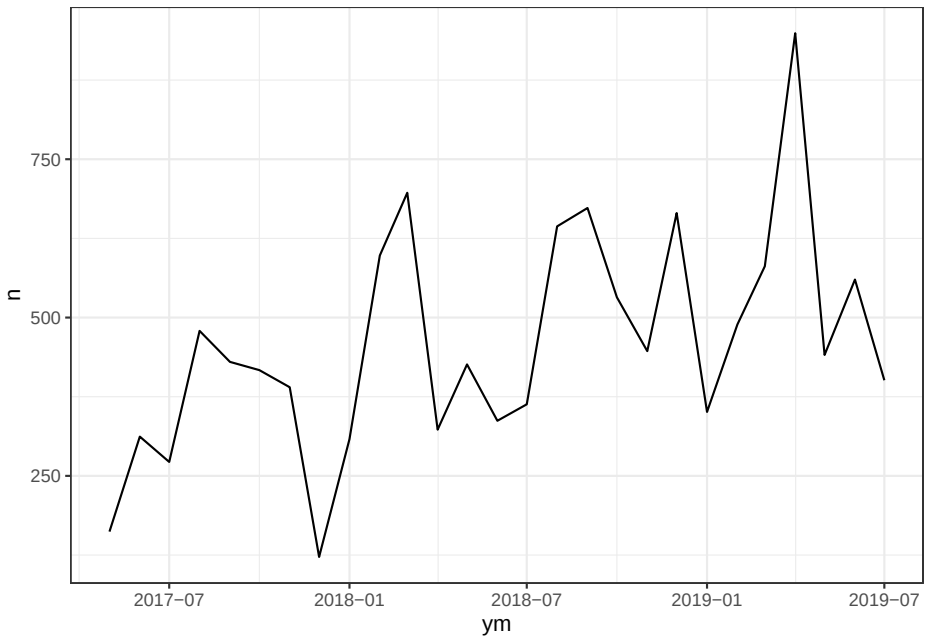
- Wir verwenden einen Ausschnitt der Daten aus der Beispielstudie. Konkret handelt es sich um Posts mit dem Suchwort *impf*, die zwischen dem 1. Mai 2016 und dem 8. Juli 2019 im Elternforum Urbia veröffentlicht wurden. Ausgeschlossen wurden unter anderem
  - sehr kurze Posts (weniger als 19 Wörter)
  - Posts mit dem Wort *schimpf*
  - Posts zur Impfung von Haustieren (nach einem kurzen Diktionär)
- Die Dokumentation zur Studie gibt weitere Informationen zur Erhebung und Bereinigung der Rohdaten.
- Diese Daten können aus Copyright- und Privacy-Gründen nicht auf GitHub veröffentlicht werden. Ich habe Sie daher im LMS hochgeladen. Bitte ladet die ZIP-Datei herunter.
  - Wenn ihr sie mit dem Code aus dem Repository integrieren wollt, müsst ihr sie in den Ordner "data" unter "R" entpacken.

```
# Laden der Daten
d = read_rds("R/data/example_data.rds")
d %>%
  print(n = 5)
```

```
## # A tibble: 12,369 x 5
##   post                          author   postdate      wc thread_title
##   <chr>                         <chr>    <date>     <int> <chr>
## 1 Wenn Impfungen zu Todesfäll~ zwerg-b~ 2018-04-06    26 HPV-Impfung
## 2 Hallo Moni Danke für deine ~ Inaktiv  2017-06-03    21 Warum so oft Scheidenp~
```

```
## 3 Hallo ja sind glaube ich dr~ danerl   2017-06-05    42 Warum so oft Scheidenp~
## 4 Guten Morgen, gibt es hier ~ butterf~ 2017-05-14   133 Impfung Deutschland/Ös~
## 5 In Österreich wird im 3., 5~ butterf~ 2017-05-15    68 Impfung Deutschland/Ös~
## # ... with 1.236e+04 more rows
```

```
d %>%
  mutate(ym = round_date(postdate, "month")) %>%
  count(ym) %>%
  ggplot(aes(ym, n)) + geom_line()
```




```
d %>%
  pull("wc") %>%
  summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      20      37      60      85     101    2493
```

- Der Datensatz besteht aus 12,369 Posts.
  - Die Variable `post` enthält den vollen Text des Posts.
  - Die Variable `author` enthält den Accountnamen, von dem der Post abgegeben wurde.
  - Die Variable `date` enthält den Tag der Veröffentlichung.
  - Die Variable `wc` enthält die Zahl der Wörter des Posts.
  - Die Variable `thread_title` enthält den Titel des Diskussions-Threads.
- Pro Monat sind zwischen ca. 120 und 1.000 Posts in unserer Stichprobe.
- Typische Posts haben einen Umfang von zwischen 40 und 100 Wörtern

(Zur Erinnerung: Sehr kurze Post wurden bereits ausgeschlossen).

2.2 Aufbereitung für das Schätzen der Topic Models

- Grundsätzlich gilt: Die verschiedenen Schritte bei der Aufbereitung des Text-Korpus kann die Ergebnisse wesentlich beeinflussen (Denny and Spirling, 2018; Maier et al., 2018). Aber ist es häufig sehr schwierig, theoretisch informierte Entscheidungen zu treffen, da
  - unsere Theorien fast immer zu vage sind, um etwas über konkret, manifeste Eigenschaften der Texte auszusagen
  - es schwer ist, die Folge einer Entscheidung für das technische Schätzen der Modelle und für die substanzielle Interpretation der Ergebnisse vorherzusagen,
  - Entscheidungen *post hoc* auf Basis der Ergebnisse wissenschaftstheoretisch und -praktisch problematisch sein können (*overfitting*, *harking* bzw. *hindsight bias*, etc.).
- In der zugrunde liegenden Studie habe ich versucht, diese Entscheidungen *a priori* zu treffen. Die Entscheidungen basieren aber zugegebenermaßen mehr auf vagen Vermutungen und für mich plausiblen und pragmatischen Überlegungen als auf einer konsistenten Theorie.
  - Entfernen von Stoppwörtern: Stoppwörter sind Wörter, die in einer Sprache häufig vorkommen und nicht wesentlich zur Bedeutung eines Texts beitragen. Hier habe ich auf Basis der deutschen Liste im Paket `stopwords` und der Worthäufigkeiten im Korpus eine Liste erstellt. Durch das *Pruning* der Dokument-Feature-Matrix ist die Auswahl der Stoppwörter aber weniger entscheidend, da Wörter, die in sehr vielen Texten des Korpus vorkommen, ohnehin entfernt werden.
  - Zusätzliche Berücksichtigung von Bi- und Tri-Grammen: Ich habe die Kombinationen von zwei oder drei Wörtern, die häufig im Korpus vorkamen, daraufhin gesichtet, ob sie für das Thema Impfen und gesundheitsrelevante Diskussionen zusätzliche Informationen enthalten, die jedes einzelne Wort alleine nicht enthält. Diese Kombinationen wurden als zusätzliche Features aufgenommen.
  - Der Argumentation und den empirischen Ergebnissen von Schofield and Mimno (2016) (deren Aufsatz übrigens einen großartigen Titel hat, großer NLP Nerd Humor) folgend habe ich auf Stemming oder Lemmatisierung verzichtet. In der Tat zeigt sich, dass Wörter mit dem gleichen Wortstamm, wie von Schofield and Mimno (2016) beschrieben, häufig im selben Topic landen.
  - Üblichen Standards (z.B. Maier et al., 2018) folgend habe ich alle Wörter in Kleinschreibung umgewandelt, Satzzeichen entfernt und URL entfernt. Zahlen habe ich beibehalten, da sie (wie die Ergebnisse auch zeigen) typische Merkmale bestimmter Perspektiven auf das

Thema Impfen sind.
  - Da wir auch an der Veränderung der Topic-Häufigkeiten über die Zeit interessiert sind, wird die Variable mit dem Erscheinungstags des Posts in eine numerische Variable umgewandelt. Sie ist so skaliert, dass der aktuellste Post den Wert 0 hat. Diese Variable können wir dann als Prädiktor beim Schätzen des *Structural Topic Model* berücksichtigen.
  - Unter *Pruning* versteht man das Entfernen von Features, die entweder in sehr weniger oder in sehr vielen Dokumenten vorkommen. Dadurch können die Größe des Datensatzes und in der Folge die zum Schätzen der Modelle nötigen Ressourcen wesentlich reduziert werden. Inhaltlich sollte das Entfernen dieser Features wenig ändern: Features, die in sehr vielen Dokumenten vorkommen, tragen nicht zur Differenzierung zwischen den Dokumenten bei. Features, die nur in sehr wenigen Dokumenten vorkommen, tragen nicht zur Definition von Topics bei, da diese durch das regelmäßige *gemeinsame* Vorkommen in Dokumenten identifiziert werden. Siehe ausführlich Maier et al. (2018).

- Die Vorbereitung des Korpus und der Dokument-Feature-Matrix erfolgte mit Funktionen aus `{quanteda}`.
  - Mit der Funktion `corpus()` wird der Datensatz in einen Text-Korpus umgewandelt. In diesem Zuge wird auch die numerische Datums-Variable erstellt. Die Variable mit dem Text des Posts duplizieren wir, damit sie zusätzlich als Meta-Datum für jeden Text gespeichert wird. Das wird später hilfreich sein, wenn wir die Ergebnisse einer Modellschätzung explorieren.
  - `custom_stopwords` und `relevant_ngrams` zeigen die Stoppwörter und Wortkombinationen, die ausgeschlossen bzw. einbezogen werden. Letztere werden mit der Funktion `dictionary()` aus `{quanteda}` erstellt.
  - Mit der Funktion `dfm()` wird der Korpus in eine Dokument-Feature-Matrix umgewandelt. Dabei werden die Standard-Schritte der Textaufbereitung durchgeführt. Sie besteht aus 12,369 Posts in den Zeilen und 41,385 Features in den Spalten. In jeder Zelle ist angegeben, wie häufig ein Feature in einem Dokument vorkommt.
  - Mit der Funktion `dfm_trim()` wird das Pruning durchgeführt. Dabei werden alle Features, die in weniger als 0.5% oder mehr als 99% der Posts vorkommen, entfernt. Nach dem Pruning enthält die Matrix nur noch 1,150 Features.
  - Zuletzt muss die Matrix in das von `stm()` benötigte Format konvertiert werden. Dabei werden zwei Posts gelöscht, die nach der Bereinigung kein einziges Feature mehr enthalten. Wichtig für den Bericht der Fallzahl in einer Publikation!

```
# Erstellen des Korpus
crps = d %>%
```

```
  mutate(txt = post, # Duplizieren des Post-Texts für Meta-Daten
         date_num = as.numeric(postdate) - max(as.numeric(postdate))) %>% # Numerische Datumsvar
  corpus(text_field = "post") # Erstellen des Korpus
crps %>%
  summary(n = 1)
```

```
## Corpus consisting of 12369 documents, showing 1 document:
##
##   Text Types Tokens Sentences       author   postdate wc thread_title
##  text1    27     29         2 zwerg-bayern 2018-04-06 26  HPV-Impfung
##
##  Wenn Impfungen zu Todesfälle oder starken Nebenwirkungen in einzelnen Fällen führen, spricht
##  date_num
##      -457
```

```
# Stoppwörter
custom_stopwords = c("ab", "aber", "ach", "all", "alle", "allem", "allen", "aller", "alles", "als

# Kombinationen von Wörtern
relevant_ngrams = dictionary(list(
  "trotz_impfung" = "trotz impfung",
  "grippe_impfen" = "grippe impfen",
  "mmr_impfung" = "mmr impfung",
  "hepatitis_b" = "hepatitis b",
  "gut_vertragen" = "gut vertragen",
  "6fach_impfung" = "6fach impfung",
  "6_fach" = "6 fach",
  "6_fach_impfung" = "6 fach impfung",
  "meningokokken_b" = "meningokokken b",
  "gute_besserung" = "gute besserung",
  "6-fach_impfung" = "6-fach impfung",
  "erhöhte_temperatur" = "erhöhte temperatur",
  "kein_fieber" = "kein fieber",
  "kein_problem" = "kein problem",
  "keine_ahnung" = "keine ahnung",
  "keine_impfung" = "keine impfung",
  "nicht_geimpft" = "nicht geimpft",
  "nicht_impfen" = "nicht impfen",
  "nicht_zu_impfen" = "nicht zu impfen",
  "selbst_entscheiden" = "selbst entscheiden"
))

# Erstellen einer Dokument-Feature-Matrix aus dem Korpus
impf_dfm = crps %>%
  dfm(stem = FALSE, tolower = TRUE, remove_punct = TRUE,
      remove = custom_stopwords,
```

```
      remove_url = TRUE, verbose = TRUE,
      thesaurus = relevant_ngrams)
```

```
## Creating a dfm from a corpus input...
##    ... lowercasing
##    ... found 12,369 documents, 41,651 features
##    ... applying a dictionary consisting of 20 keys
##    ... removed 286 features
```

```
impf_dfm
```

```
## Document-feature matrix of: 12,369 documents, 41,385 features (99.9% sparse) and 6 d
##        features
## docs    TROTZ_IMPFUNG GRIPPE_IMPFEN MMR_IMPFUNG HEPATITIS_B GUT_VERTRAGEN
##   text1             0             0           0           0             0
##   text2             0             0           0           0             0
##   text3             0             0           0           0             0
##   text4             1             0           0           0             0
##   text5             0             0           0           0             0
##   text6             0             0           0           0             0
##        features
## docs    6FACH_IMPFUNG 6_FACH 6_FACH_IMPFUNG MENINGOKOKKEN_B GUTE_BESSERUNG
##   text1             0      0              0               0              0
##   text2             0      0              0               0              0
##   text3             0      0              0               0              0
##   text4             1      0              0               0              0
##   text5             0      0              0               0              0
##   text6             0      0              0               0              0
## [ reached max_ndoc ... 12,363 more documents, reached max_nfeat ... 41,375 more feat
```

```
# Pruning
impf_dfm = impf_dfm %>%
  dfm_trim(max_docfreq = 0.99, min_docfreq = 0.005, docfreq_type = "prop")
impf_dfm
```

```
## Document-feature matrix of: 12,369 documents, 1,150 features (98.2% sparse) and 6 do
##        features
## docs    TROTZ_IMPFUNG GRIPPE_IMPFEN MMR_IMPFUNG GUT_VERTRAGEN 6_FACH
##   text1             0             0           0             0      0
##   text2             0             0           0             0      0
##   text3             0             0           0             0      0
##   text4             1             0           0             0      0
##   text5             0             0           0             0      0
##   text6             0             0           0             0      0
##        features
```

```
## docs    MENINGOKOKKEN_B GUTE_BESSERUNG ERHÖHTE_TEMPERATUR KEIN_FIEBER
##   text1               0              0                   0           0
##   text2               0              0                   0           0
##   text3               0              0                   0           0
##   text4               0              0                   0           0
##   text5               0              0                   0           0
##   text6               0              0                   0           0
##        features
## docs    KEIN_PROBLEM
##   text1            0
##   text2            0
##   text3            0
##   text4            0
##   text5            0
##   text6            0
## [ reached max_ndoc ... 12,363 more documents, reached max_nfeat ... 1,140 more features ]
```

```
# Überblick: Die häufigsten Features im Korpus
impf_dfm %>%
  colSums() %>%
  enframe() %>%
  arrange(desc(value)) %>%
  slice(1:20) %>%
  kable()
```

name	value
impfung	4519
impfen	4356
kind	3690
lassen	3257
immer	2705
mehr	2594
kinder	2503
gibt	2184
geimpft	2176
impfungen	2174
gut	2115
hallo	1898
einfach	1767
lg	1720
2	1704
bekommen	1585
ganz	1584
erst	1558
geht	1492
arzt	1411

```
# als (beliebte, wenn auch nur mittel informative) Wordcloud
impf_dfm %>%
  textplot_wordcloud()
```

```
## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <98>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <8a>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <98>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <8a>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Fontmetrik ist für das Unicode-Zeichen U+1f60a unbekannt

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <98>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <82>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <98>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <82>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Fontmetrik ist für das Unicode-Zeichen U+1f602 unbekannt

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <98>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <89>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <98>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <89>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Fontmetrik ist für das Unicode-Zeichen U+1f609 unbekannt

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <88>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>

## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <99>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <88>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Fontmetrik ist für das Unicode-Zeichen U+1f648 unbekannt
```

```
## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>
```

```
## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>
```

```
## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <98>
```

```
## Warning in graphics::strwidth(word[i], cex = size[i]): Konvertierungsfehler für
## ' ' in 'mbcsToSbcs': Punkt ersetzt <85>
```

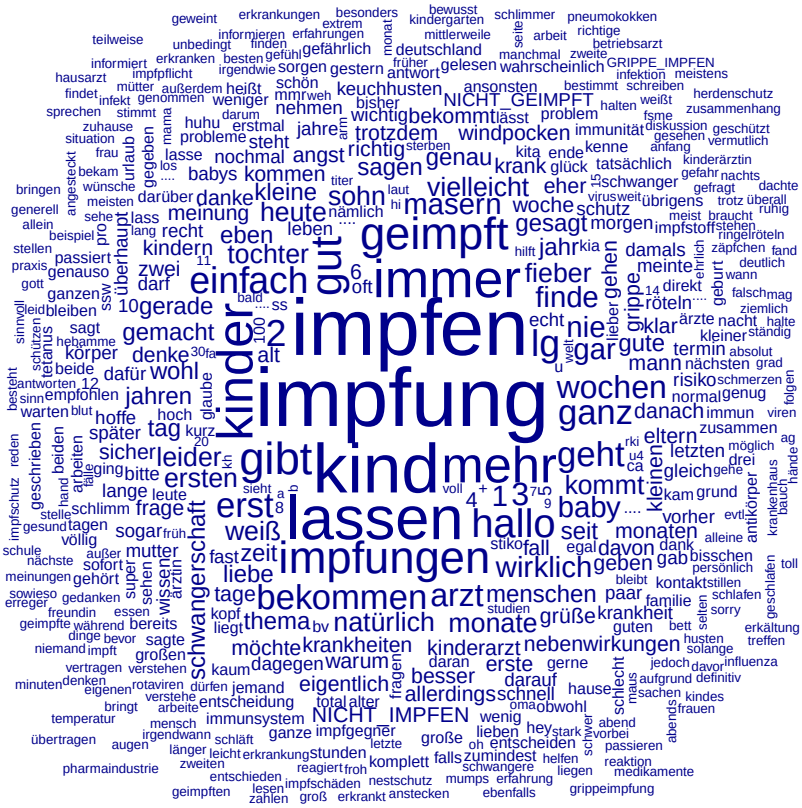
```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <f0>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <9f>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <98>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Konvertierungsfehler für ' ' in 'mbcsToSbcs': Punkt ersetzt <85>
```

```
## Warning in text.default(x1, y1, word[i], cex = (1 + adjust) * size[i], offset =
## 0, : Fontmetrik ist für das Unicode-Zeichen U+1f605 unbekannt
```




```
# Umwandeln der dfm in das Format für stm
impf_stm = impf_dfm %>%
  quanteda::convert(to = "stm")
```

```
## Warning in dfm2stm(x, docvars, omit_empty = TRUE): Dropped empty document(s):
## text12225, text12271
```

Chapter 3

Modellspezifikation, Modellvergleich und Modellauswahl

3.1 Modellspezifikation

- Die Funktion `stm()` aus dem gleichnamigen Paket bietet zahlreiche Möglichkeiten, Details der Modellspezifikation und -schätzung anzupassen. Wir beschränken uns im Folgenden auf drei wesentliche Einstellungen:
  - `K`: Die Zahl der Topics.
  - `prevalence`: Eine Formel zur Vorhersage der Topic-Prävalenzen
  - `init.type`: Wie soll der Startpunkt für die Modellschätzung gewählt werden?
- Zu weiteren Details siehe für einen Überblick `?stm` und Roberts et al. (2019) für eine ausführliche Erläuterung.
- Eine Modell-Spezifikation könnte z.B. so aussehen:

```
modelfit = stm(documents = impf_stm$documents,
               vocab = impf_stm$vocab,
               data = impf_stm$meta,
               K = 10,
               prevalence = ~s(date_num),
               init.type = "Spectral")
```

- Mit den ersten drei Inputs übergeben wir die Daten aus der im letzten Abschnitt erstellten Dokument-Feature-Matrix.

- Mit `K` geben wir an, wie viele Themen es geben soll. Mit dieser Syntax würde ein Modell mit $k = 10$ Themen geschätzt. Wie wir bei der Wahl eines geeigneten k vorgehen können, ist Thema des folgenden Unterabschnitts.
- Mit der Formel zu `prevalence` geben wir an, welche Dokument-Variablen mit dem Auftreten der Topics zusammenhängen.
  - In der Formel wird die abhängige Variable vor der Tilde (~) frei gelassen. Es wird immer der Zusammenhang mit dem Auftreten von allen k Topics geschätzt. In diesem Beispiel schätzen wir, wie sich das Auftreten der Topics über den Untersuchungszeitraum hinweg verändert. Details zum Schätzen von Zusammenhängen mit Kovariaten folgen später in diesem Workshop.
  - Mit `init.type` wird angegeben, wie `stm()` die Ausgangswerte für die Modellschätzung bestimmen soll. Die Default-Einstellung ist "Spectral". Ich empfehle diese Einstellung aus folgenden Gründen:
    * Sie ist deterministisch, d.h., sie führt gegeben derselben Daten und desselben Modells immer zu derselben Lösung. So wird die Reproduzierbarkeit sichergestellt.
    * Sie ist effizient, d.h., dass von diesem Startpunkt aus relativ schnell die finale Lösung gefunden wird.
    * Wenn eine andere Einstellung für die Ausgangswerte gewählt wird, müssen mehrere Schätzungen für eine Spezifikation durchgeführt werden. Nur so kann geprüft werden, ob die Ausgangswerte das Ergebnis beeinflussen.
- Allgemein muss beachtet werden, dass die Schätzung eines Structural Topic Model mit `stm()` trotz der Effizienz der Implementierung sehr rechenintensiv ist. Die Schätzung des oben beschriebenen Modells dauert auf meinem recht leistungsfähigen Notebook bereits ca. eine Minute. Es empfiehlt sich daher, die Modelle immer in neue Objekte zu speichern und diese ggf. direkt auf der Festplatte zu sichern. Um die Berechnungszeiten in diesem Workshop kurz zu halten, stelle ich die Ergebnisse der Modellschätzungen über das LMS zur Verfügung. Wenn ihr diese herunterladet und in den Ordner "data" kopiert, muss das Modell nicht neu geschätzt werden.

3.2 Modellvergleich

- Eine zentrale Frage ist die Wahl eines geeigneten k, also der Zahl von Topics, die in den Dokumenten identifiziert werden sollen. Wichtig ist zuerst die Feststellung, dass es in der angewandten Analyse kein *per se* richtiges oder falsches k gibt.
- Wie viele Topics nützlich sind, hängt von Umfang von Zusammensetzung des Materials und vom substantiellen Forschungsinteresse ab.
- Um ein geeignetes k zu finden, gehen wir in der Regel modellvergleichend vor. Wir schätzen Modelle mit unterschiedlich vielen Topics und prüfen

dann, welche Modelle besser zu den Daten und zum Forschungsinteresse passen.

- Hinweise für einen allgemeinen Ausgangspunkt, in welchem Bereich nützliche k zu finden sein könnten, liefert die Paket-Hilfe:

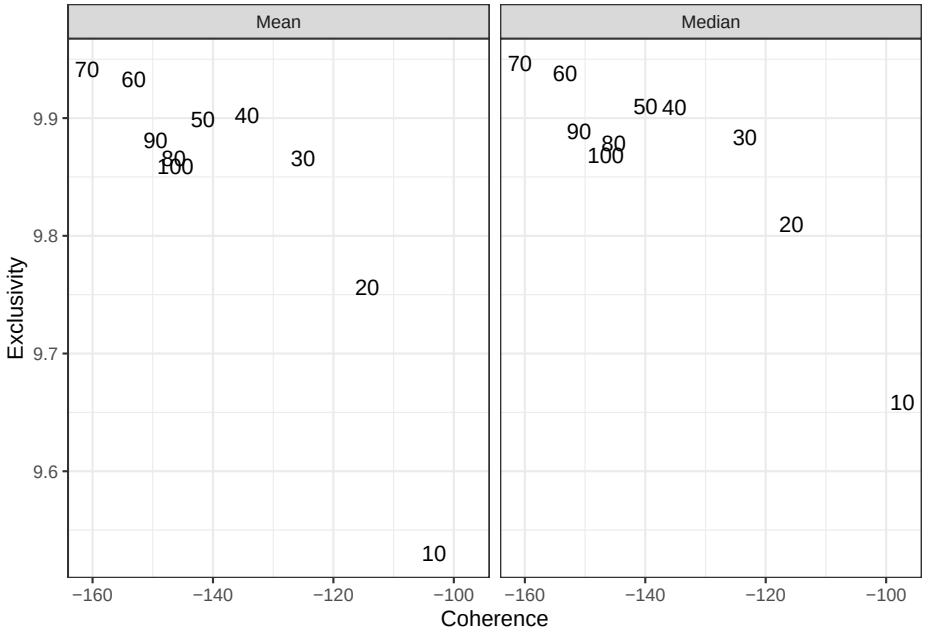
  > The most important user input in parametric topic models is the number of topics. There is no right answer to the appropriate number of topics. More topics will give more fine-grained representations of the data at the potential cost of being less precisely estimated. […] For short corpora focused on very specific subject matter (such as survey experiments) 3-10 topics is a useful starting range. For small corpora (a few hundred to a few thousand) 5-50 topics is a good place to start. Beyond these rough guidelines it is application specific. Previous applications in political science with medium sized corpora (10k to 100k documents) have found 60-100 topics to work well. For larger corpora 100 topics is a useful default size. Of course, your mileage may vary. — ?stm

- Hier werden zwei wichtige Kriterien, die unser Nachdenken über die Spannweite von zu Berücksichtigten k leiten können, deutlich:
  - Quantität des Materials: Je mehr Dokumente, desto mehr Topics.
  - Varianz im Inhalt: Je mehr inhaltliche Varianz, desto mehr Topics (an einem Beispiel: für 10k Nachrichtenbeiträge aus dem Wirtschaftsteil brauchen wir weniger Topics als für 10k Nachrichtenbeiträge, die aus allen Ressorts kommen).
- Im vorliegenden Fall haben wir einen kleinen bis mittleren Korpus (ca. 13k Dokumente, die größtenteils recht kurz sind). Wir können von einer mittleren inhaltlichen Varianz ausgehen. Einerseits haben wir Posts bewusst danach ausgewählt, dass sie sich mit dem Thema Impfen beschäftigen, was die Varianz einschränkt. Andererseits wissen wir, dass in Online-Foren die verschiedensten Perspektiven auf dieses Thema vorkommen können, was für Varianz sorgt.
- Wir gehen im Folgenden in mehreren Schritten vor:
  1) Um eine allgemeine Orientierung zu erhalten, in welcher Range Modelle zu finden sind, die gut zu den Daten passen, schätzen wir 10 Modelle von $k = 10$ bis $k = 100$ mit einem Abstand von jeweils 10 Topics. Diese Modelle vergleichen wir anhand von einigen statistischen Maßen, um die Zahl der Kandidatenmodelle einzuschränken.
  2) Wir interpretieren die besten Modelle substantiell und entscheiden, welche Topic-Anzahl für das Forschungsinteresse hilfreicher scheint.
  3) Wir schätzen weitere Modelle zwischen den besten Modellen aus 2). Wir prüfen, wie sich die Topics verändern. Zudem achten wir darauf, ob bei Modellen mit größeren k interessante Topics hinzukommen oder ob sich mehr Ambivalenzen zeigen.
  4) Wir entscheiden uns für ein Modell. Dieses beschreiben wir dann ausführlich.
- Da das Schätzen der Modelle recht lange dauert, parallelisieren wir die

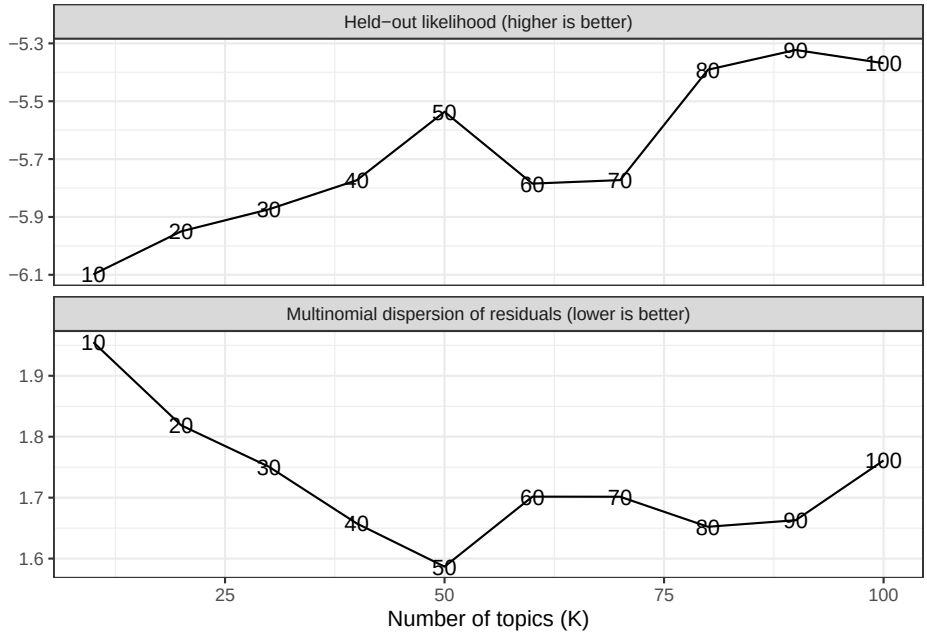
Berechnung. Dazu nutze ich das Paket `furrr`. Es sei an dieser Stelle darauf hingewiesen, dass das Paket vor allem unter Windows mit RStudio für Probleme sorgen kann. Es ist daher empfehlenswert, das Skript zum Schätzen und Speichern der Modelle in der Konsole oder im Terminal auszuführen. Noch schneller geht es für diesen Workshop, die bereits geschätzten Modelle aus dem LMS zu laden.

```
# 1) Modelle mit K = 10, ..., K = 100
# Modelle schätzen bzw. laden
if (file.exists("R/data/models10_100.rds")) {
# Schneller: Modelle aus LMS laden
  many_models = read_rds("R/data/models10_100.rds")
} else {
# Vorsicht: Schätzen dauert auf meinem MacBook Pro 2020 i9 32 GB RAM 10 Minuten
  Ks = seq(10, 100, by = 10)
  tic()
  plan(multiprocess(workers = 10))
  many_models = tibble(K = Ks) %>%
    mutate(topic_model = future_map(K, ~stm(documents = impf_stm$documents,
                                            vocab = impf_stm$vocab,
                                            data = impf_stm$meta,
                                            init.type = "Spectral",
                                            K = ., verbose = FALSE),
                                    .progress = TRUE))
  plan(sequential)
  toc()
  saveRDS(many_models, "R/data/models10_100.rds")
}

# Quantitative Indikatoren der Modellqualität berechnen
# Inspired by https://juliasilge.com/blog/evaluating-stm/
if (file.exists("R/data/eval10_100.rds")) {
  # Schneller: Modellevaluation aus LMS laden
  model_eval = read_rds("R/data/eval10_100.rds")
} else {
  # Evaluation dauert auf meinem MacBook Pro 2020 i9 32 GB RAM 24 Sekunden
  tic()
  heldout = make.heldout(documents = impf_stm$documents, vocab = impf_stm$vocab)
  plan(multiprocess(workers = 10))
  model_eval = many_models %>%
    mutate(exclusivity = future_map(topic_model, exclusivity),
           semantic_coherence = future_map(topic_model, semanticCoherence, impf_stm$do
           eval_heldout = future_map(topic_model, eval.heldout, heldout$missing),
           residual = future_map(topic_model, checkResiduals, impf_stm$documents),
           residuals = future_map_dbl(residual, "dispersion"),
           held_out_likelihood = future_map_dbl(eval_heldout, "expected.heldout")) %>%
```

```
    select(-topic_model)
  plan(sequential)
  toc()
  saveRDS(model_eval, "R/data/eval10_100.rds")
}

# Exklusivität und Semantische Kohärenz (Mean, Median)
model_eval %>%
  select(K, Exclusivity = exclusivity, Coherence = semantic_coherence) %>%
  mutate_at(-1, .funs = list(Mean = ~map_dbl(.x, mean), Median = ~map_dbl(.x, median))) %>%
  select_if(negate(is.list)) %>%
  gather(metric, value, -K) %>%
  separate(metric, c("measure", "metric"), "_") %>%
  spread(measure, value) %>%
  ggplot(aes(Coherence, Exclusivity, label = K)) + geom_text() + facet_wrap("metric")
```




```
# Held-out-likelihood und multinomiale Residuen
model_eval %>%
  select(K, `Held-out likelihood (higher is better)` = held_out_likelihood, `Multinomial dispersi
  gather(measure, value, -K) %>%
  ggplot(aes(K, value, label = K)) + geom_line() + geom_text() + facet_wrap("measure", scales = '
```




```
# Speichern der Modelle mit 30, 40 und 50 Topics für qualitative Analyse
# Benennung und Datenstruktur für stminsights
# Auskommentiert, da bereits im LMS gespeichert
out = impf_stm
m30 = many_models$topic_model[[3]]
m40 = many_models$topic_model[[4]]
m50 = many_models$topic_model[[5]]
# e30 = estimateEffect(formula = 1:30~1, stmobj = m30, metadata = impf_stm$meta)
# e40 = estimateEffect(formula = 1:40~1, stmobj = m40, metadata = impf_stm$meta)
# e50 = estimateEffect(formula = 1:50~1, stmobj = m50, metadata = impf_stm$meta)
save(out, m30, m40, m50, file = "R/data/models30_50.rdata")
# save(out, m30, m40, m50, e30, e40, e50, file = "R/data/models30_50.rdata")
```

Bibliography

Denny, M. J. and Spirling, A. (2018). Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It. *Political Analysis*, 26(2):168–189.

Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., Pfetsch, B., Heyer, G., Reber, U., Häussler, T., Schmid-Petri, H., and Adam, S. (2018). Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. *Communication Methods and Measures*, 12(2-3):93–118.

Roberts, M. E., Stewart, B. M., and Tingley, D. (2019). Stm: An R Package for Structural Topic Models. *Journal of Statistical Software*, 91(1):1–40.

Schofield, A. and Mimno, D. (2016). Comparing apples to apple: The effects of stemmers on topic models. *Transactions of the Association for Computational Linguistics*, 4:287–300.