Universität Münster

Quantitative Textanalyse

Sitzung 7: Datenaufbereitung – Vom Datensatz zum Textmodell

Mirko Wegemann

Universität Münster
Institut für Politikwissenschaft

20. November 2024

Logistik

- in den letzten Wochen haben wir uns mit verschiedenen Möglichkeiten der Datenerschließung beschäftigt (verschiedene Formen von Scraping, APIs)
- in dieser Woche lernen wir, wie wir Textdateien im Rohformat in verarbeitende Textdateien konvertieren

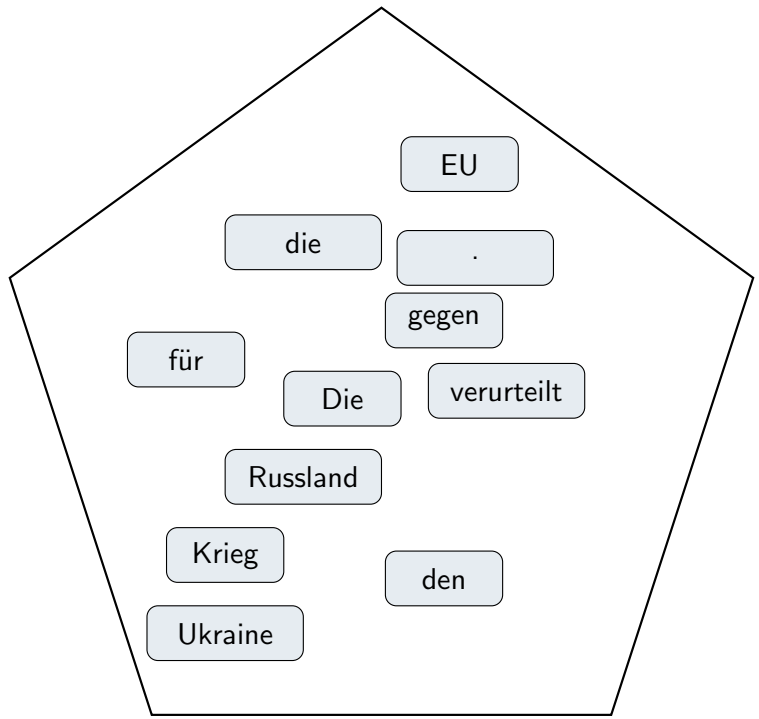
Bags-of-words I

Zur Erinnerung: Text ist komplex und für unsere quantitativen Modelle nicht nutzbar.
Dementsprechend müssen wir ihn zunächst in eine Repräsentation von Zahlen umwandeln.

Bags-of-words II

Bags-of-words (BoW) erstellt für jedes Dokument *D* eines Korpus einen Vektor, welcher die Token *t* enthält.

Die EU verurteilt Russland für den Krieg gegen die Ukraine.



Bags-of-words III

Die generelle Idee dahinter ist, dass wir die **Bedeutung** von Text durch das genutzte **Vokabular** nachvollziehen können. Ein Vergleich zwischen Dokumenten d_1 und d_2 findet alleinig auf Basis der **Häufigkeiten von Tokens** statt.

Bags-of-words IV

Document D1	*The child makes the dog happy* the: 2, dog: 1, makes: 1, child: 1, happy: 1
Document D2	*The dog makes the child happy* the: 2, child: 1, makes: 1, dog: 1, happy: 1

↓

	child	dog	happy	makes	the	**BoW Vector representations**
D1	1	1	1	1	2	**[1,1,1,1,2]**
D2	1	1	1	1	2	**[1,1,1,1,2]**

Two documents with different meanings, yet same BoW representation (Source: AIML.com Research)

Datenvorbereitung in R

Außer für deskriptive Zwecke verwenden wir unseren Rohdatensatz meist nicht direkt, sondern müssen ihn transformieren.
Dabei gibt es drei Transformationsschritte:

1. Von ‘data.frame‘ zu einem Corpus-Objekt
2. Vom corpus zu einem tokens-Objekt
3. Von Tokens zu einer **D**ocument **F**requency **M**atrix (auch **D**ocument-**T**erm-**M**atrix genannt)

Ein Corpus

Ein **Corpus**-Objekt behandelt jede Zeile eines Datensatzes als ein Dokument.

```
> corp <- corpus(df$sentence)
> head(corp,2)
Corpus bestehend aus 2 Dokumenten.
text1 :
"Peter Costello, Chris McDiven, my parliamentary
    colleagues a..."

text2 :
"This campaign, more than any other that I have
    been involved..."
```

Ein Corpus II

- ...ist in der Regel nur der erste Transformationsschritt
- ...gibt uns darüber hinaus einen ersten Überblick über die Daten mit `summary(corpus_object)`
- ...wird für einige `quanteda`-Funktionen verwendet; wie den Lesbarkeitsindex `textstat_readability(corpus_object)`
- ...mehr zu Corpus-Objekten

Ein Tokens-Objekt

Bei der Tokenisierung führen wir den noch zusammenhängenden Text in sogenannte Tokens über (z. B. Sätze, Wörter, Zeichen) und geben uns eine Matrix in folgender Form aus:

```
> toks <- tokens(corp, what="word")
>
> head(toks[[1]], 20)
[1] "Peter"          "Costello"       ","
                "Chris"          "McDiven"
    ","              "my"
    "parliamentary"
[9] "colleagues"     "and"            "my"
                "fellow"         "Australians"    "."
```

Ein Tokens-Objekt I

- Tokens-Objekte erlauben viele Vorverarbeitungsschritte wie `remove_punct=T`, `remove_numbers=T`, `remove_symbols=T`, `remove_url=T`.
- mit `tokens_remove(stopwords())` können wir Stopwörter aus einem Objekt entfernen (wie "und", "der", "ein", "in").
- `kwic(token_object, "economy", window=n)` zeigt, welche Begriffe ein Wort umgeben.
- es gibt verschiedene Tokenizer: eine leistungsstärkere Alternative zu quanteda ist `library(spacyR)`, das Wörter lemmatisiert und Wortarten (z. B. Nomen, Verb, Name) erkennt.
- ...mehr zu Token-Objekten

Eine Document-Frequency-Matrix

Eine *dfm* ist eine eine Matrix, bei der jede Zeile ein Dokument und jede Spalte ein Feature ist. Jede Zelle zeigt, wie oft ein Feature in einem Dokument verwendet wurde.

```
> m_dfm <- dfm(toks, tolower = T)
>
> m_dfm
Document-feature matrix of: 187,689 documents,
    44,409 features (99.96% sparse) and 0 docvars.
features
docs    peter costello  ,    chris    mcdiven
text1       1         1    2        1          1
text2       0         0    2        0          0
[ reached max_ndoc ... 187,683 more documents,
    reached max_nfeat ... 44,399 more features ]
```

Eine Document-Frequency-Matrix I

dfm-Objekte werden für die meisten unserer Analysen verwendet.

- mit `docvars(dfm, "name") <- df$name` können wir Metainformationen hinzufügen (z. B. zu Redner*in, Zeit, Thema).
- für ein dfm-Objekt sind mehrere Funktionen zum text pre-processing verfügbar:
  - `dfm_keep`, `dfm_remove`, `dfm_subset`, `dfm_trim` und `dfm_replace` erlauben das Entfernen oder Ändern von Merkmalen (Begriffen) in der Matrix
  - `dfm_lookup` erlaubt das Anwenden eines Wörterbuchs
  - `dfm_group` gruppiert die dfm anhand einer in docvars() definierten Gruppe
  - `dfm_tfidf` gewichtet jedes Token anhand einer term-frequency-inverse-document-matrix

Eine Document-Frequency-Matrix II

- `dfm_sample` nimmt eine zufällige Stichprobe von Dokumenten (oft für Trainings-/Testsplits in Klassifikationsaufgaben verwendet)
- `dfm_match` vergleicht Merkmale der dfm mit einer anderen dfm und erstellt dieselbe Struktur

- **...mehr zu dfm-Objekten**

Und jetzt in R

Warum komprimieren?

Wir arbeiten oftmals mit großen Datensätzen. Aus Effizienz, aber auch substantiellen Interessen, möchten wir daher teils einige Features aus unserem Textkorpus entfernen. Wir stehen dabei meist vor einem Zwiespalt:

- Informationsverlust
- Große/wenig informative Datenstruktur

Verschiedene Möglichkeiten der Datenkomprimierung I

Einige der Features, welche wir löschen bzw. verändern sind...

1. Satzzeichen
2. Zahlen
3. Umwandlung auf Kleinschreibung
4. Stemming/Lemmatisierung
5. Stopwords
6. Nutzung von n-grams
7. Entfernen von Stopwords

Wenn wir all diese Verfahren zusammennehmen, kommen wir auf $2^7 = 128$ verschiedene Kombinationen.

Probleme beim Komprimieren I

Was wir optimalerweise benötigen, ist eine **Theorie über unseren Textkorpus**. Aus welchen Gründen können welche Features ausgeschlossen werden.

Das bedeutet, wir sollten unseren **Datensatz gut kennen**. Leider ist dies nicht immer der Fall und Forscher*innen folgen einfach etablierten Verfahren, um den Korpus zu komprimieren.

Probleme beim Komprimieren II

Denny and Spirling (2017, September) beschreiben die Probleme solch eines Vorgehens

- in **supervised** Verfahren können wir die Performance unserer Modelle gut evaluieren; die Sinnhaftigkeit von pre-processing Verfahren kann daher gut analysiert werden
- in **unsupervised** Verfahren ist dies schwieriger; die Modelle sind sehr sensitiv zum Input, den wir geben; welches Verfahren besser ist als die anderen, ist schwierig zu entscheiden

Probleme beim Komprimieren III

Specification	Most Left	Most Right
P-N-S-W-3-I	Lab $_{1983}$	Cons $_{1983}$
N-S-W-3	Lab $_{1987}$	Cons $_{1987}$
N-L-3	Lab $_{1992}$	Cons $_{1987}$
N-L-S	Lab $_{1983}$	Cons $_{1992}$

Table 3: Some example specifications which differ in terms of the manifestos they place on the (far) left and (far) right under the Wordfish model.

Figure: Vier verschiedene Pre-Processing Verfahren; vier verschiedene Ergebnisse (Denny and Spirling 2017, September, p. 16)

Probleme beim Komprimieren IV

Ihre Lösung:

- **Grundlage**: jedes Dokument hat eine gewisse Distanz zu anderen Dokumenten [basierend auf Gemeinsamkeiten der Features]; wir können alle Dokumente nach ihrer Distanz ranken
- **Frage**: wie sehr verändert sich das Ranking zwischen den Dokumenten, wenn eine bestimmte Pre-Processing Abfolge genutzt wird
- **Intuition**: je mehr sich das Ranking ändert, desto größer ist der Einfluss des Pre-Processings auf unsere folgenden Modelle
- **Handlungsempfehlung**: Wenn wir keinen guten Grund haben, einflussreiche Pre-Processing Schritte durchzuführen, dann sollten wir es lieber lassen

Descriptive statistics and visualization I

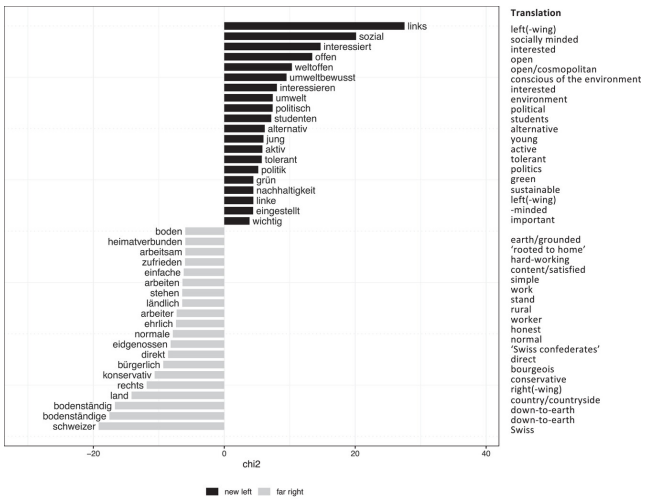
Nachdem wir all diese Schritte durchgeführt haben, können wir

unsere Daten visualisieren:

Bauer et al. (2017); in `quanteda`: `textplot_wordcloud(dfm)`

Descriptive statistics and visualization II

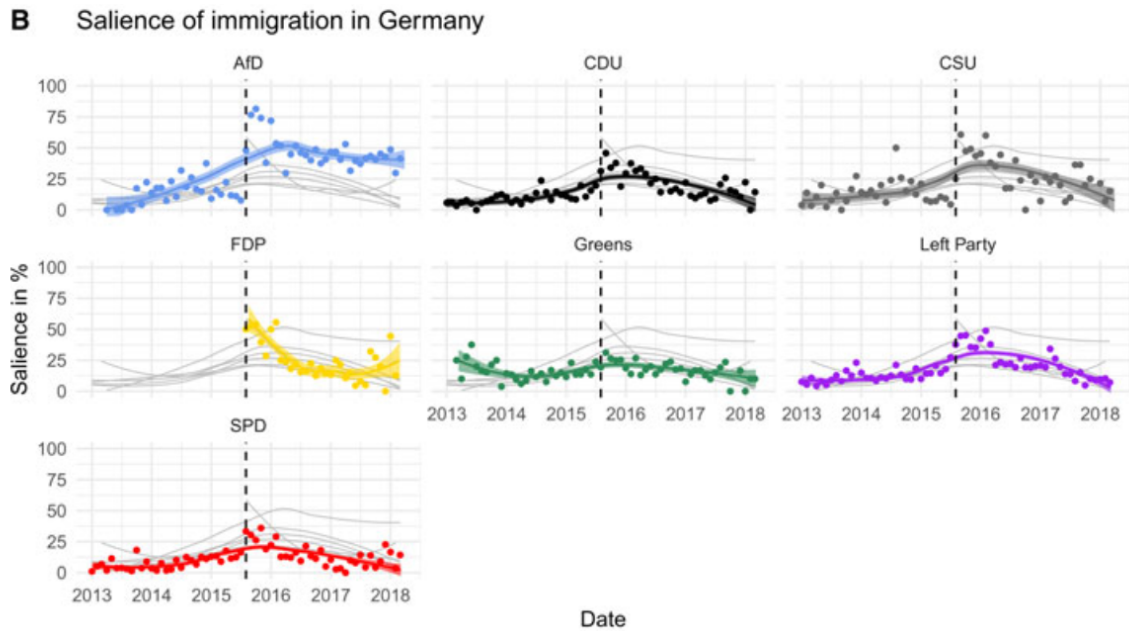
Wir können Statistiken wie die sogenannte *keyness*-Statistik berechnen (basiert auf χ^2), welche zeigt, welche Begriffe besonders mit bestimmten Kommunikator*innen verbunden sind.



Zollinger (2024); in `quanteda`: `textstat_keyness()` und `textplot_keyness()`

Descriptive statistics and visualization III

Zudem können wir auch einen Dictionary-Task durchführen.



Gessler and Hunger (2022); in `quanteda`: `dfm_lookup(dfm, dict)`

Und jetzt in R

Was haben wir heute gelernt...

- unsere Textdateien müssen zunächst in numerische Vektoren umgewandelt werden
- für *bags-of-words* Modelle setzen wir darauf auf Document-Feature-Matrizen
- wir können unsere Datensätze weiter komprimieren, sollten dafür aber gute Gründe haben

Ausblick

- in der nächsten Woche wenden wir das erste Mal eine Textanalyse an; wir beschäftigen uns zunächst mit *unsupervised topic models*
- Literatur:
  1. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003).Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022
  2. mehr Hintergrund zu structural topic models: Roberts, M. E., Stewart, B. M., & Tingley, D. (2019).Stm: An R Package for Structural Topic Models. *Journal of Statistical Software*, *91*, 1–40. https://doi.org/10.18637/jss.v091.i02

Literatur I

Bauer, P. C., Barberá, P., Ackermann, K., & Venetz, A. (2017).Is the Left-Right Scale a Valid Measure of Ideology? *Political Behavior*, *39*(3), 553–583. https://doi.org/10.1007/s11109-016-9368-2

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003).Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022.

Denny, M., & Spirling, A. (2017, September). Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It. https://doi.org/10.2139/ssrn.2849145

Gessler, T., & Hunger, S. (2022).How the Refugee Crisis and Radical Right Parties Shape Party Competition on Immigration. *Political Science Research and Methods*, *10*, 524–544. https://doi.org/10.1017/psrm.2021.64

Literatur II

Roberts, M. E., Stewart, B. M., & Tingley, D. (2019).Stm: An R Package for Structural Topic Models. *Journal of Statistical Software*, *91*, 1–40. https://doi.org/10.18637/jss.v091.i02

Zollinger, D. (2024).Cleavage Identities in Voters' Own Words: Harnessing Open-Ended Survey Responses. *American Journal of Political Science*, *68*(1), 139–159. https://doi.org/10.1111/ajps.12743