Universität Münster

Text as Data

Session 7: Data preparation

Mirko Wegemann

Universität Münster
Institut für Politikwissenschaft

10 June 2026

Our plan for today

- in the last two weeks we have learned different ways to scrape data from the web and transform it into a structured format
- this week we will learn how to convert raw text files into processable text files
- before that: presentation by Beyza

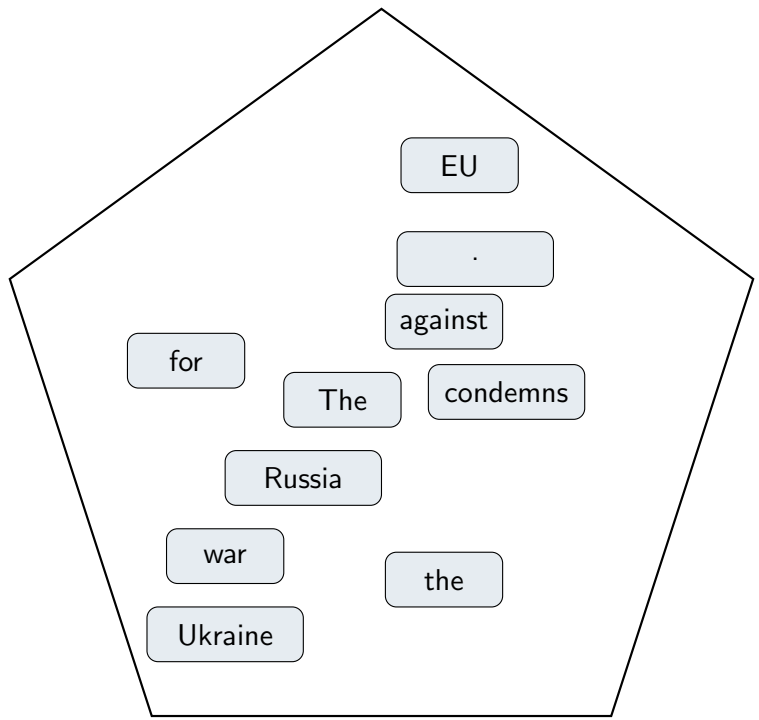
Bags-of-words I

Recall: Text is complex and not directly usable for our quantitative models. Therefore, we need to convert it into a numerical representation.

Bags-of-words II

Bags-of-words (BoW) creates a vector for each document D in a corpus that contains the tokens t.

The EU condemns Russia for the war against Ukraine.



Bags-of-words III

The general idea behind this is that we can understand the **meaning** of text through the **vocabulary** used. A comparison between documents d_1 and d_2 is based solely on the **frequencies of tokens**.

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)

Data Preparation in R

We usually do not use our raw dataset directly for descriptive purposes, but need to transform it.
During this process, there are three transformation steps:

1. From ‘data.frame‘ to a Corpus object
2. From corpus to a tokens object
3. From Tokens to a **D**ocument **F**requency **M**atrix (also known as **D**ocument-**T**erm-**M**atrix)

A Corpus

A **Corpus**-object treats each row of a dataset as a document.

```
1  > corp <- corpus(df$sentence)
2  > head(corp,2)
3  Corpus comprised of 2 documents.
4    text1 :
5      "Peter Costello, Chris McDiven, my parliamentary
           colleagues a..."
6
7    text2 :
8      "This campaign, more than any other that I have
           been involved..."
```

A Corpus II

- ...is usually only the first transformation step.
- ...also provides us with an initial overview of the data using `summary(corpus_object)`.
- ...is used for some `quanteda`-functions, such as the readability index `textstat_readability(corpus_object)`.
- ...more about Corpus objects

A Tokens Object

During tokenization, we break down the still connected text into so-called tokens (e.g., sentences, words, characters) and output a matrix in the following form:

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

A Tokens Object I

- Tokens objects allow for many preprocessing steps such as `remove_punct=T`, `remove_numbers=T`, `remove_symbols=T`, `remove_url=T`.
- With `tokens_remove(stopwords())`, we can remove stopwords from an object (such as "and", "the", "a", "in").
- `kwic(token_object, "economy", window=n)` shows which words surround a given word.
- There are different tokenizers: a more powerful alternative to quanteda is `library(spacyR)`, which lemmatizes words and recognizes parts of speech (such as nouns, verbs, names).
- ...more about Token objects

A Document-Frequency Matrix

A *dfm* is a matrix where each row represents a document and each column represents a feature. Each cell shows how often a feature is used in a document.

```
> 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 ]
```

A Document-Frequency Matrix

dfm objects are used for most of our analyses.

- With `docvars(dfm, "name") <- df$name`, we can add metadata (such as speaker, time, topic).
- For a dfm object, several text preprocessing functions are available:
  - `dfm_keep`, `dfm_remove`, `dfm_subset`, `dfm_trim`, and `dfm_replace` allow the removal or modification of features (terms) in the matrix.
  - `dfm_lookup` allows the application of a dictionary.
  - `dfm_group` groups the dfm based on a group defined by `docvars()`.
  - `dfm_tfidf` weights each token based on a term-frequency-inverse-document-matrix.
  - `dfm_sample` takes a random sample of documents (often used for training/test splits in classification tasks).
  - `dfm_match` compares features of the dfm with another dfm and creates the same structure.
  - more about dfm objects

And now in R

Why Compress?

We often work with large datasets. For efficiency, as well as substantive reasons, we therefore want to remove some features from our text corpus partially. We are usually faced with a trade-off:

- Information loss
- Large/insubstantial data structure

Various Methods of Data Compression I

Some of the features we can remove or modify include...

1. Punctuation
2. Numbers
3. Converting to lowercase
4. Stemming/Lemmatization
5. Stopwords
6. Using n-grams
7. infrequently used terms

When we combine all these methods, we arrive at $2^7 = 128$ different combinations (+ there are infinite choices to made regarding stopwords, n-grams and infrequently used terms).

Problems with Compression I

What we ideally need is a **theory about our text corpus** to understand which features can be excluded for specific reasons.

This means that we should have a good understanding of our **dataset**. Unfortunately, this is not always the case, and researchers often follow established procedures to compress the corpus.

Problems with Compression II

Denny and Spirling (2017, September) describe the problems with such an approach:

- In **supervised** methods, we can evaluate the performance of our models well; thus, the significance of preprocessing techniques can be analyzed.
- In **unsupervised** methods, this is more difficult; the models are very sensitive to the input provided. It is hard to decide which method is better than the others.

Problems with Compression 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: Four different preprocessing methods; four different outcomes (Denny and Spirling 2017, September, p. 16)

Problems with Compression IV

Your Solution:

- **Foundation**: Each document has a certain distance to other documents [based on the similarities of features]; we can rank all documents by their distance.
- **Question**: How much does the ranking of documents change when a specific preprocessing sequence is used?
- **Intuition**: The more the ranking changes, the greater the impact of preprocessing on our subsequent models.
- **Action Recommendation**: If we have no good reason to perform influential preprocessing steps, it is better to leave them out.

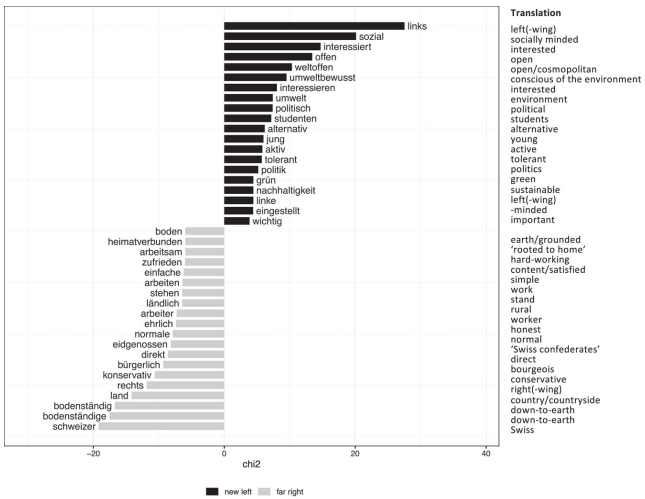
Descriptive statistics and visualization I

After performing all these steps, we can visualize our data:

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

Descriptive statistics and visualization II

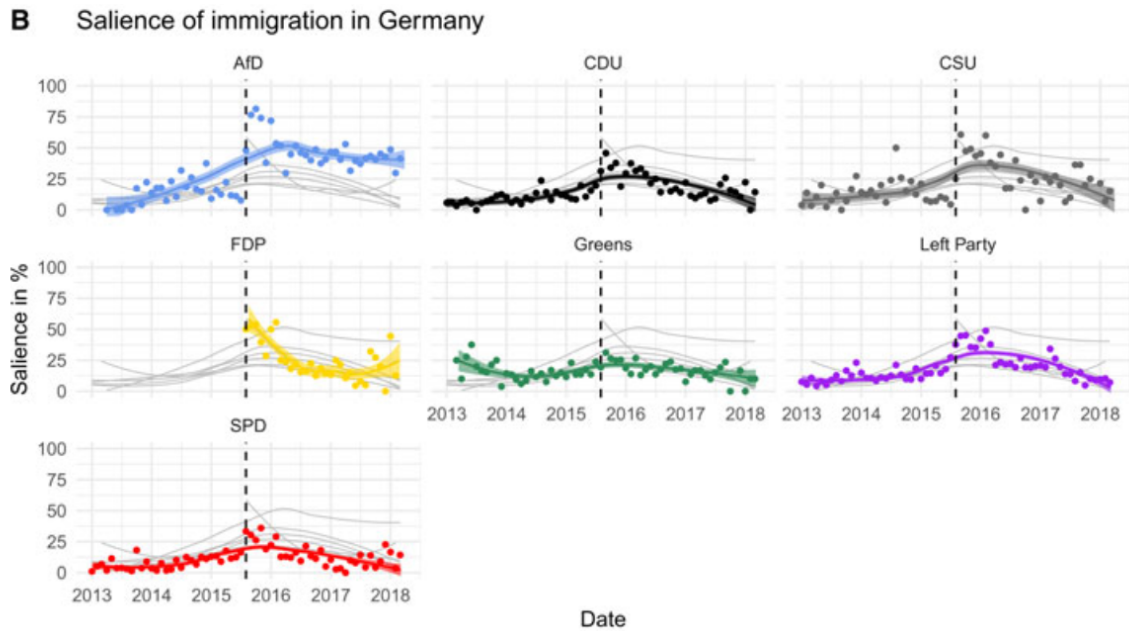
We can also calculate statistics such as the keyness statistic (based on χ^2), which shows which terms are particularly associated with certain communicators.



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

Descriptive statistics and visualization III

Moreover, we can also perform a dictionary task.



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

And now in R

What Have We Learned Today...

- Our text files must first be converted into numerical vectors.
- For *bag-of-words* models, we rely on document-feature matrices.
- We can further compress our datasets, but we should have good reasons for doing so.

Next Week

- Next week, we will (finally) apply a text analysis, introducing *unsupervised topic models*.
- Literature:
  1. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003).Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*(null), 993–1022
  2. More background on 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

Literature 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

Literature 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