LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Session 4:
Topic Modeling

Agenda

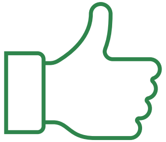
1. Introduction to Topic Modeling
2. Deciding on Parameter Settings
3. Interpretation
4. Test against Quality Criteria
5. Outro

How can I access materials?

- Materials include
  - Reading list
  - Slides
  - R/Python Code (via Google Colab Notebook)

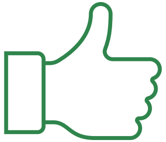
- Access via:
  - Summer school website
  - GitHub

Expectation management for this session

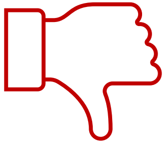


- Know how topic modeling works, including methodological steps and how (not) to use them
- Run a topic model

Expectation management for this session



- Know how topic modeling works, including methodological steps and how (not) to use them
- Run a topic model



- Understand each line of code
- Run a „**good**“ topic model (limited input on hyperparameters, finding K, validation, etc.)

Preparing the hands-on part in R/Python

https://github.com/valeriehase/Salamanca-CSS-SummerSchool

Folder: "Topic modeling"

1_Slides_Topic modeling: Contains slides for respective session

2_Reading list_Topic modeling: Contains reading list for respective session. Please make sure to read the required text before the respective session.

3_R Code_Topic modeling: Contains R code for respective session via Colab Notebook. Open in Colab

4_Python Code_Topic modeling: Contains Python code for respective session via Colab Notebook. Open in Colab

data_tvseries : Contains CSV-dataset on best rated TV series. Provided under MIT license via Kaggle.

```python
import pandas as pd
import numpy as np
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
import gensim
from gensim import corpora
from gensim import matutils
from gensim.models import LdaModel
from gensim.models import CoherenceModel
import matplotlib.pyplot as plt

# Ensure you have the necessary NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
```

Preparing the hands-on part in R/Python

R

```r
# Install relevant packages
install.packages("dplyr")
install.packages("RCurl")
install.packages("quanteda")
install.packages("stm")
install.packages("reshape2")
install.packages("ggplot2")
```

```r
# We activate relevant packages
library("dplyr")
library("RCurl")
library("quanteda")
library("stm")
library("reshape2")
library("ggplot2")
```

Python

```python
import pandas as pd
import numpy as np
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
import gensim
from gensim import corpora
from gensim import matutils
from gensim.models import LdaModel
from gensim.models import CoherenceModel
import matplotlib.pyplot as plt

# Ensure you have the necessary NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
```

1. Introduction to Topic Modeling

Remember cluster analysis?

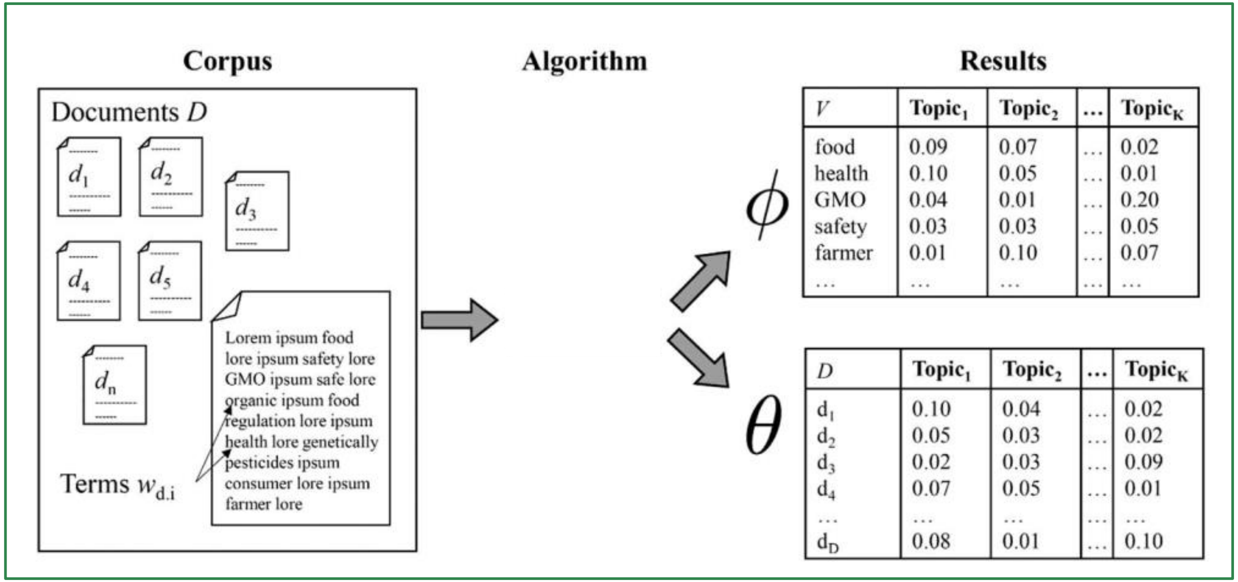
- We want to find overarching patterns/types in data
- We are interested in an exploratory analysis where categories are unknown beforehand
- Topic modeling works (somewhat) similar…

Topic modeling: Definition

„computational content-analysis technique […] used to investigate the "hidden" thematic structure of […] texts" (Maier et al., 2018, p. 93)

- Method: Unsupervised machine learning (ML) technique
- Approach: **identify previously unknown latent topics** based on frequently co-occurring manifest words.

Topic modeling: Definition



V	$Topic_1$	$Topic_2$	...	$Topic_K$
food	0.09	0.07	...	0.02
health	0.10	0.05	...	0.01
GMO	0.04	0.01	...	0.20
safety	0.03	0.03	...	0.05
farmer	0.01	0.10	...	0.07
...	...	...	...	...

D	$Topic_1$	$Topic_2$	...	$Topic_K$
d_1	0.10	0.04	...	0.02
d_2	0.05	0.03	...	0.02
d_3	0.02	0.03	...	0.09
d_4	0.07	0.05	...	0.01
...	...	...	...	...
d_D	0.08	0.01	...	0.10

(Maier et al., 2018, p. 94)

Topic modeling: Definition

- Probabilistic mixed-membership model:
  - Each feature has non-zero probability for each topic (*ϕ-matrix*)
  - Each topic has non-zero probability for each document (*θ-matrix*)
- Generative model: find best fitting model for generating our corpus of documents
  - joint-probability distribution of observed variables (features *i* in documents *d*) & latent variables (*ϕ, θ*)

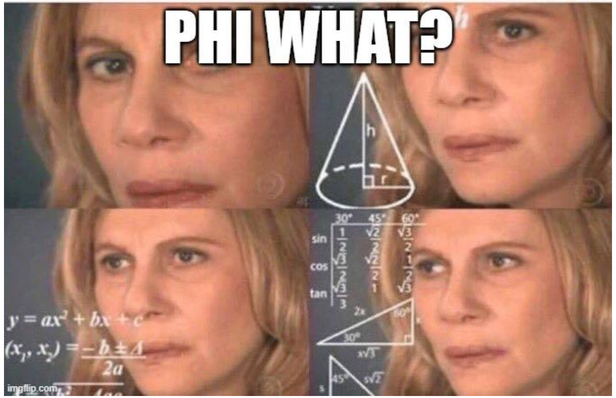
Two central „results“

Word-topic matrix:

- conditional probability of features being prevalent in topics
- used to generate word lists to describe topics (“top features”)

V	Topic_1	Topic_2	…	Topic_K
food	0.09	0.07	…	0.02
health	0.10	0.05	…	0.01
GMO	0.04	0.01	…	0.20
safety	0.03	0.03	…	0.05
farmer	0.01	0.10	…	0.07
…	…	…	…	…

(Maier et al., 2018, p. 94)



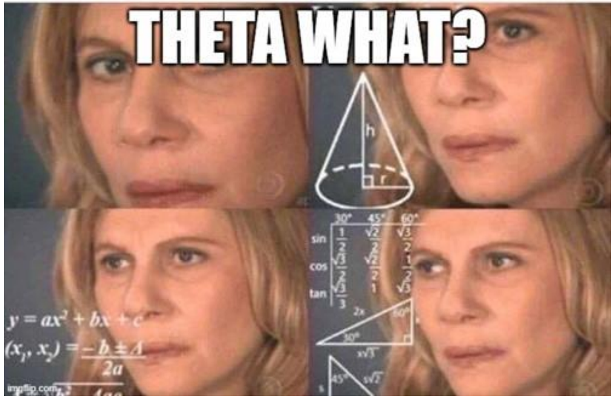
Two central „results“

Document-topic matrix:

- conditional probability of topics being prevalent in documents
- used to generate document lists to describe topics (“top documents”)

D	$Topic_1$	$Topic_2$	...	$Topic_K$
d_1	0.10	0.04	...	0.02
d_2	0.05	0.03	...	0.02
d_3	0.02	0.03	...	0.09
d_4	0.07	0.05	...	0.01
...	...	...	...	...
d_D	0.08	0.01	...	0.10

(Maier et al., 2018, p. 94)



Key methodological steps

- Preprocessing (see session 2)
    - Settings can affect results (Denny & Spirling, 2018; Maier et al., 2020)

Key methodological steps

- Preprocessing (see session 2)
- Deciding on parameter settings
  - Algorithm (Churchill et al., 2020; Eshima et al., 2023; Roberts et al., 2014)
  - K as number of topics
  - α as prior for θ
  - β as prior for ϕ

Key methodological steps

- Preprocessing (see session 2)
- Deciding on parameter settings
- Interpretation

Key methodological steps

- Preprocessing (see session 2)
- Deciding on parameter settings
- Interpretation
- Test against Quality Criteria (Bernhard et al., 2023; Quinn et al., 2010)

Example study

Global Environmental Change 70 (2021) 102353

Contents lists available at ScienceDirect

Global Environmental Change

journal homepage: www.elsevier.com/locate/gloenvcha

ELSEVIER

Climate change in news media across the globe: An automated analysis of issue attention and themes in climate change coverage in 10 countries (2006–2018)

Valerie Hase[a,*], Daniela Mahl[a], Mike S. Schäfer[a], Tobias R. Keller[b]

[a] Department of Communication and Media Research, University of Zurich, Switzerland
[b] gfs.bern, Bern, Switzerland

ARTICLE INFO

Keywords:
Climate Change
Media Coverage
Longitudinal Analysis
Computational Social Science
Automated Content Analysis
Comparative Research

ABSTRACT

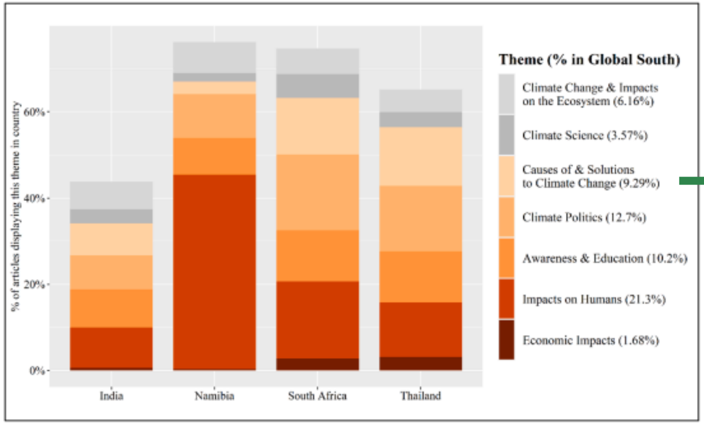
Climate change poses a challenge to countries across the world, with news media being an important source of information on the issue. To understand how and how much news media cover climate change, this study compares coverage in ten countries from the Global North and the Global South between 2006 and 2018 (N = 71,674). Based on a panel analysis, we illustrate that news media attention varies across countries and is often associated with political, scientific, and (partly) societal focusing events. Based on an automated content analysis, we also find that news media do not only cover ecological changes or climate science, but that they focus predominantly on the societal dimension of climate change: They emphasize how humans are aware of, affected by, battle, or cause climate change. Overall, the study illustrates important differences between the Global North and the Global South. While countries from the Global North cover climate change more frequently, countries from the Global South focus more on its challenges and implications for society at large, i.e., the societal dimension of climate change.

Example study

Sample 10 countries*, 2006-2018 (N = 71,674 articles)

*Australia, Canada, Germany, India, Namibia, New Zealand, South Africa, Thailand, UK, USA

Method Structural topic modeling



Societal Dimension (43.63%)

Theme: Causes of & Solutions to Climate Change (13.83%)

Topic: Carbon-related incentives & policies	2.75%	carbon price, carbon tax, trading
Topic: Clean energy	2.05%	nuclear power, solar, nuclear energy
Topic: Divestment	1.19%	investors, investment, divestment
Topic: Energy efficiency	1.12%	efficiency, efficient, heating
Topic: Oil drilling & fracking	1.09%	oil, fracking, shell

Key take-aways

- **Topic modeling**: unsupervised ML approach to identify previously unknown latent topics based on frequently co-occurring manifest words.
- **Key steps**:
  - Preprocessing
  - Deciding on parameter settings
  - Interpretation
  - Test against quality criteria

Key take-aways

- **Topic modeling**: unsupervised ML approach to identify previously unknown latent topics based on frequently co-occurring manifest words.
- **Key steps**:
  - Preprocessing
  - Deciding on parameter settings: **the number of topics K**
  - Interpretation
  - Test against quality criteria

2. Deciding on Parameter Settings

Deciding on parameter settings

- Among other parameters (α, β), researchers have to specify the number of topics K
- No single best solution; decision often quite subjective.
- You could rely on…
  - Statistical fit (e.g., coherence, perplexity)
  - Interpretability (e.g., Top features of topics)
  - Rank-1 metric (e.g., frequent vs. infrequent topics)

Deciding on parameter settings

- Among other parameters (α, β), researchers have to specify the number of topics K
- No single best solution; decision often quite subjective.
- You could rely on…
  - **Statistical fit** (e.g., coherence, perplexity)
  - **Interpretability** (e.g., Top features of topics)
  - Rank-1 metric (e.g., frequent vs. infrequent topics)

→ Let us test solutions with K = 4 vs. K = 6 topics!

Time for R/Python!

Getting text into R/Python

R

```r
# We load data (a csv-file with ratings and content of TV series) from the Github repository
url = getURL("https://raw.githubusercontent.com/valeriehase/Salamanca-CSS-SummerSchool/
main/Processing%20text%20and%20text%20as%20data/data_tvseries.csv")
data = read.csv2(text = url)
```

Python

```python
# We load data (a csv-file with ratings and content of TV series) from the Github repository
url = "https://raw.githubusercontent.com/valeriehase/Salamanca-CSS-SummerSchool/
main/Processing%20text%20and%20text%20as%20data/data_tvseries.csv"
data = pd.read_csv(url, sep = ";")
```

Preprocessing

R

```
tokens <- tokens(data$Description,
                 what = "word",
                 remove_punct = TRUE,
                 remove_numbers = TRUE) %>%
          tokens_tolower() %>%
          tokens_remove(stopwords("english")) %>%
          tokens_wordstem()

#Additional steps: apply relative pruning
dfm <- dfm_trim(dfm(tokens), min_docfreq = 0.005, max_docfreq = 0.99,
                docfreq_type = "prop", verbose = TRUE)
```

Python

```
stop_words = set(stopwords.words("english"))
stemmer = PorterStemmer()

#Preprocess
def clean_description_dfm(description):
    # Tokenize the description
    words = word_tokenize(description)
    # Remove special signs and convert to lower case
    words = [word.lower() for word in words if word.isalpha()]
    # Remove stopwords
    words = [word for word in words if word not in stop_words]
    # Apply stemming
    words = [stemmer.stem(word) for word in words]
    #Additionally re-join as string
    return ' '.join(words)  # Join the tokens back into a single string

tokens_dfm = [clean_description_dfm(description) for description in data["Description"]]

#Create a document-feature matrix, with relative pruning
vectorizer = CountVectorizer(min_df = 0.004, max_df = .99)
dfm = vectorizer.fit_transform(tokens_dfm)
```

Check bag-of-words representation (in Python)

```
#Check result
pd.DataFrame(dfm.todense(), columns = vectorizer.get_feature_names_out()).head()
```

	abil	accid	across	act	action	activ	actor	adapt	adolesc	adult	...	women	work
0	0	0	0	0	0	0	0	0	0	0	...	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0

5 rows × 755 columns

Deciding on K: Statistical fit

R

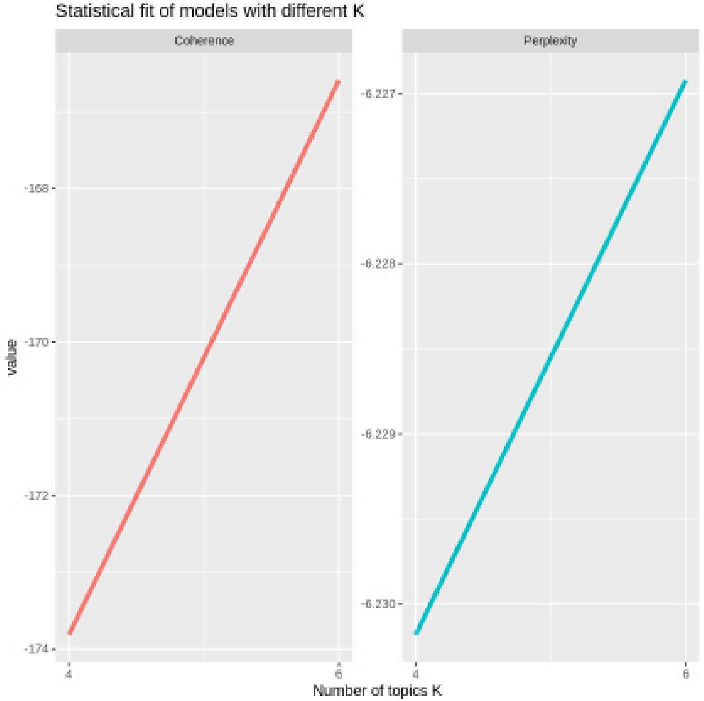
```
#Transform to right format for stm package
dfm stm <- convert(dfm, to = "stm")
K <- c(4,6)
fit <- searchK(dfm_stm$documents, dfm_stm$vocab, K = K, verbose = TRUE)

# Create graph
plot <- data.frame("K" = K,
                   "Coherence" = unlist(fit$results$semcoh),
                   "Perplexity" = unlist(fit$results$heldout))

# Reshape to long format
plot <- melt(plot, id = c("K"))

# Create graph
plot <- data.frame("K" = K,
                   "Coherence" = unlist(fit$results$semcoh),
                   "Perplexity" = unlist(fit$results$heldout))

# Reshape to long format
plot <- melt(plot, id = c("K"))

#Plot result
ggplot(plot, aes(K, value, color = variable)) +
  geom_line(linewidth = 1.5, show.legend = FALSE) +
  scale_x_continuous(breaks = c(4, 6)) +
  facet_wrap(~ variable, scales = "free_y") +
  labs(x = "Number of topics K",
       title = "Statistical fit of models with different K")
```

Deciding on K: Statistical fit

Python

```
corpus = matutils.Sparse2Corpus(dfm, documents_columns = False)
dictionary = dict(enumerate(vectorizer.get_feature_names_out()))

result = []
for k in [4,6 ]:
    m = LdaModel(
        corpus,
        num_topics = k,
        id2word = dictionary,
        random_state = 2024,
    )
    perplexity = m.log_perplexity(corpus)
    coherence = CoherenceModel(
        model = m, corpus = corpus, coherence = "u_mass"
    ).get_coherence()
    result.append(dict(k = k, perplexity = perplexity, coherence = coherence))

result = pd.DataFrame(result)
result.plot(x = "k", y=["perplexity", "coherence"])
plt.xticks([4, 6])
plt.show()
```

Example statistical fit (in R)



Coherence: should be high – how frequently do most probable words in topic co-occur?

Perplexity: should be low – how well does the model predict unseen documents?

Deciding on K: Interpretability

R

```
model_4K <- stm(documents = dfm_stm$documents,
         vocab = dfm_stm$vocab,
         K = 4)

model_6K <- stm(documents = dfm_stm$documents,
         vocab = dfm_stm$vocab,
         K = 6)

#for K = 4
topics_4 <- labelTopics(model_4K, n=10)
topics_4 <- data.frame("features" = t(topics_4$frex))
colnames(topics_4) <- paste("Topics", c(1:4))
topics_4

#for K = 6
topics_6 <- labelTopics(model_6K, n=10)
topics_6 <- data.frame("features" = t(topics_6$frex))
colnames(topics_6) <- paste("Topics", c(1:6))
topics_6

findThoughts(model_4K, data$Description, topics = 2 , n = 1)
```

Top features

Top documents

Deciding on K: Interpretability

Python

```
model_4K = LdaModel(corpus, num_topics = 4, id2word = dictionary, random_state = 2024)
model_6K = LdaModel(corpus, num_topics = 6, id2word = dictionary, random_state = 2024)

#for K = 4
pd.DataFrame(
    {
        f"Topic {n}": [w for (w, tw) in words]
        for (n, words) in model_4K.show_topics(formatted=False)
    }
)

#for K = 6
pd.DataFrame(
    {
        f"Topic {n}": [w for (w, tw) in words]
        for (n, words) in model_6K.show_topics(formatted=False)
    }
)
```

Top features

Deciding on K: Interpretability

Python

```
def get_representative_docs_for_topic(model, corpus, documents, topic_id, top_n = 5):
    """
    Extract the most representative documents for a specific topic in an LDA model.

    Parameters:
    - model: The trained LdaModel object.
    - corpus: The corpus used for training the LDA model.
    - documents: The original documents corresponding to the corpus.
    - topic_id: The topic ID for which to extract the most representative documents.
    - top_n: The number of most representative documents to extract for the topic.

    Returns:
    - representative_docs: A list of the most representative documents for the specified topic.
    """
    representative_docs = []

    # Iterate over each document in the corpus
    for doc_id, bow in enumerate(corpus):
        # Get the topic distribution for the document
        topic_distribution = model.get_document_topics(bow, minimum_probability=0)

        # Store the document's topic probability for the specified topic
        for tid, prob in topic_distribution:
            if tid == topic_id:
                representative_docs.append((doc_id, prob))

    # Sort the documents for the specified topic by probability in descending order
    representative_docs.sort(key=lambda x: x[1], reverse=True)
    # Keep only the top_n most representative documents
    representative_docs = representative_docs[:top_n]

    # Convert document indices to actual documents
    representative_docs = [documents[doc_id] for doc_id, _ in representative_docs]

    return representative_docs

# Get the most representative document for the 2nd topic (1st index, therefore topic_id = 1)
representative_docs_for_topic = get_representative_docs_for_topic(model = model_4K, corpus = corpus, documents = data["Description"], topic_id = 1, top_n = 1)

# Print a representative documents for the  topic
representative_docs_for_topic
```

Top documents

Your turn!

Can you...

- **Discuss** which model seems to be the better fit?
- **Explain** based on what metrics you decided this?

Key Take-Aways

- **Parameter settings**: configurations of the topic model which have to be specified before running the model
- **K**: the number of topics which researchers want to study. Oftentimes, there is no single solution:
  - Check statistical fit (often not correlated with human judgements, see Chang et al., 2009)
  - Check interpretability
  - Check Rank-1 metric

Short break

3. Interpretation

Interpreting results

- Identification & exclusion of background topics
- Interpretation & labeling of relevant topics
- “Assignment” of topics to documents
- Connection to theory (Günther, 2022)

Interpreting results

- **Identification & exclusion of background topics**
- **Interpretation & labeling of relevant topics**
- “Assignment” of topics to documents
- Connection to theory (Günther, 2022)

Time for R/Python!

Interpretation

R

```r
# Run the model you decided on
model <- stm(documents = dfm_stm$documents,
          vocab = dfm_stm$vocab,
          K = 4)

#Save top 20 features across topics and forms of weighting
labels <- labelTopics(model, n = 15)

#only keep FREX weighting
topwords <- data.frame("features" = t(labels$frex))

#assign topic number as column name
colnames(topwords) <- paste("Topics", c(1:4))

#Return the result
topwords[1:4]

# Create theta matrix
theta <- make.dt(model)

#Get most representative topics for topic 1
theta %>%
  arrange(desc(Topic1)) %>%
  head()

# Check related topic
data$Description[345]
```

Top features

Top document

Interpretation

Python

```
model = LdaModel(corpus, num_topics = 4, id2word = dictionary, random_state = 2024)

#Check top words
pd.DataFrame(
    {
        f"Topic {n}": [w for (w, tw) in words]
        for (n, words) in model_4K.show_topics(formatted=False)
    }
)


# Get the most representative document for first topic
get_representative_docs_for_topic(model = model_4K, corpus = corpus,
                                  documents = data["Description"], topic_id = 0, top_n = 1)
```

Top features

Top document

Your turn!

Can you…

- **Identify** relevant vs. background topics?
- **Find** a single „label“ for each relevant topic?
- If you have more time: Run models with **other K** and see if they fit better?

Visualizing topic proportions

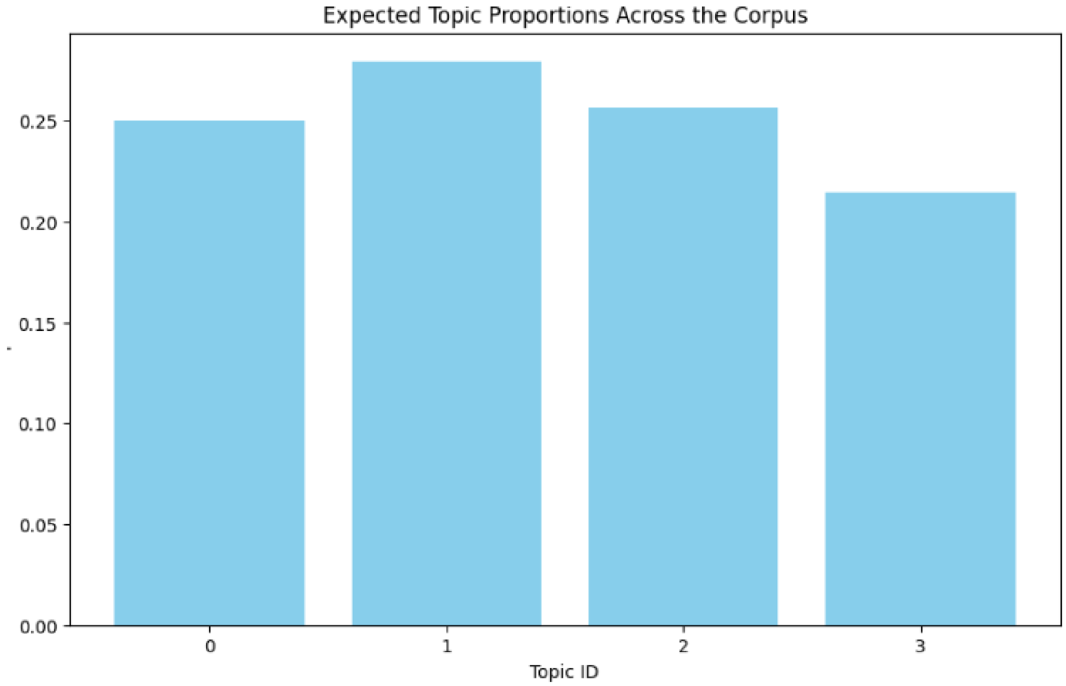
R

```r
plot(model)
```

Python

```python
# Infer topic distributions for each document
topic_distributions = [model.get_document_topics(bow, minimum_probability = 0) for bow in corpus]

# Aggregate topic proportions across the corpus
num_topics = model.num_topics
topic_proportions = np.zeros(num_topics)

for doc_topics in topic_distributions:
    for topic_id, prop in doc_topics:
        topic_proportions[topic_id] += prop

# Normalize to get proportions
topic_proportions /= len(corpus)

# Plot the topic proportions
plt.figure(figsize=(10, 6))
plt.bar(range(num_topics), topic_proportions, color='skyblue')
plt.xlabel('Topic ID')
plt.ylabel('Proportion')
plt.title('Expected Topic Proportions Across the Corpus')
plt.xticks(range(num_topics))
plt.show()
```

Visualizing topic proportions (in Python)



Key Take-Aways

- **Background topics**: Topics that are incoherent and/or represent aspects not relevant to the study at hand (e.g., type of language)
- **Top features**: features that describe a given topic
- **Top documents**: documents that describe a given topic

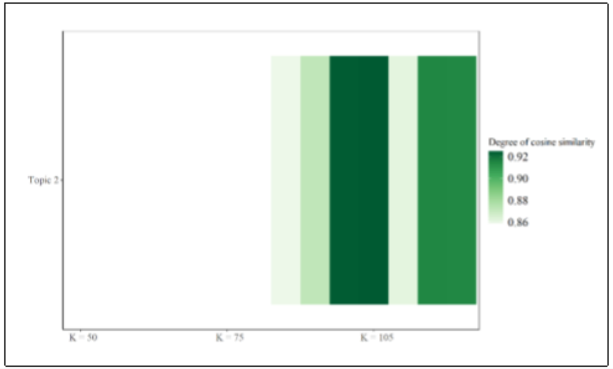
4. Test against Quality Criteria

Quality Criteria: Validity (Bernhard et al., 2023; Quinn et al., 2020)

- Compare against human annotations
- Intrusion test (features, documents) (Chan & Sältzer, 2020)
- Compare to external events
- Reflect on theoretical meaning of „topics“: but please be **very, very careful** about equating topics with frames (see critically Nicholls et al., 2021)

Example: Excluding background topics

Information on Topic 2 from the model with $K = 85$ topics

Top Terms	*Please see the following top terms:* lights, awareness, earth hour, wwf, campaign, hour, earth, switch, concert, initiative, raise, organisers, stadium, launched, bangkok, launch save, switched, switching, message
Top Documents	*Please read the following documents:* Hindu_2009-9-9_718.txt; NZHerald_2008-07-21_2538.txt; The Nation_2015-3-26_23512.txt; NZHerald_2009-03-30_1901.txt; Hindu_2008-5-3_1601.txt; Toronto Star_2008-4-17_1923.txt; The Star_2018-3-9.txt; The Sydney Morning Herald_2009-3-17_33364.txt; Hindu_2012-3-22_4051.txt ; Hindu_2009-9-8_720.txt
Rank-1 Metric	Absolute (N = Number of articles): 618 Relative (% of articles in corpus): .86%
Robustness of Topic across K	*Figure 1: Robustness of topic for different choices of K* 

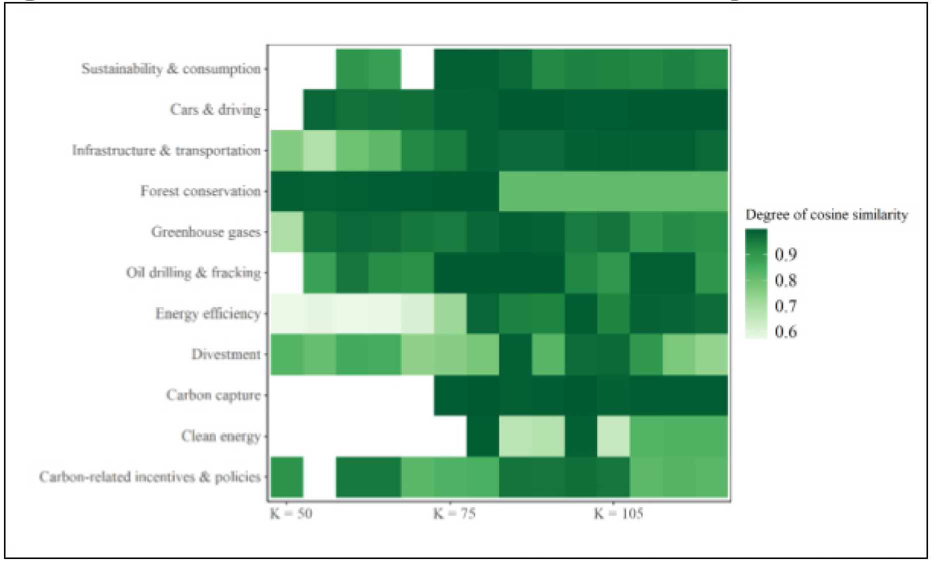
(Hase et al., 2021 – Supplement)

Quality Criteria: Robustness (Roberts et al., 2016; Wilkerson & Casas, 2017)

- Topic models can converge to local modes
- Even if you run the same code on the same computer, you may not be able to reproduce results!

Example: Robustness (Roberts et al., 2016; Wilkerson & Casas, 2017)

Figure D1. Robustness of Theme: Causes of & Solutions to Climate Change

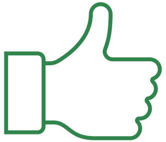


Note: Green spaces indicate that the topic in our reference model with $K = 85$ was reproduced in models with other K. Y-axis identifies topic in our reference model, x-axis identifies robustness model with different K. The darker the green, the higher the cosine similarity between top terms of topic in the reference model and the robustness models.

(Hase et al., 2021 – Supplement)

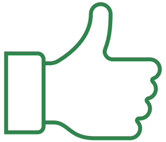
5. Take Away & Outlooks

How (not) to use topic models

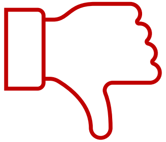


- Explore topics in large corpora
- Use to understand and potentially substructure corpus for follow-up analysis
- Combine with more qualitative methods

How (not) to use topic models



- Explore topics in large corpora
- Use to understand and potentially substructure corpus for follow-up analysis
- Combine with more qualitative methods



- Overinterpret topics: Events? Issues? Frames? (Maier et al., 2019; Nicholls et al., 2021)
- Ignore degrees of freedom for methodological decisions & their downstream effects

Thanks!
Any Questions?

Dr. Valerie Hase
IfKW, LMU Munich

orcid.org/0000-0001-6656-4894

valeriehase

@valeriehase.bsky.social

www.valerie-hase.com

References

- Bernhard, J., Teuffenbach, J., & Boomgaarden, H. G. (2023). Topic Model Validation Methods and their Impact on Model Selection and Evaluation. *Computational Communication Research*, *5*(1), 1. https://doi.org/10.5117/CCR2023.1.13.BERN
- Chan, C., & Sältzer, M. (2020). oolong: An R package for validating automated content analysis tools. *Journal of Open Source Software*, *5*(55), 2461. https://doi.org/10.21105/joss.02461
- Chang, ., Gerrish, S., Wang, C., Boyd-Graber, J.-L., Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems*, 288–296.
- Chen, Y., Peng, Z., Kim, S.-H., & Choi, C. W. (2023). What We Can Do and Cannot Do with Topic Modeling: A Systematic Review. *Communication Methods and Measures*, *17*(2), 111–130. https://doi.org/10.1080/19312458.2023.2167965
- Churchill, R., & Singh, L. (2022). The Evolution of Topic Modeling. *ACM Computing Surveys*, *54*(10s), 1–35. https://doi.org/10.1145/3507900
- Denny, M. J., & 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. https://doi.org/10.1017/pan.2017.44
- Eshima, S., Imai, K., & Sasaki, T. (2023). Keyword-Assisted Topic Models. *American Journal of Political Science*, ajps.12779. https://doi.org/10.1111/ajps.12779
- Günther, E. (2022). *Topic Modeling: Algorithmische Themenkonzepte in Gegenstand und Methodik der Kommunikationswissenschaft*. Herbert von Halem Verlag.
- Hase, V., Mahl, D., Schäfer, M. S., & Keller, T. R. (2021). Climate change in news media across the globe: An automated analysis of issue attention and themes in climate change coverage in 10 countries (2006–2018). *Global Environmental Change*, *70*, 102353. https://doi.org/10.1016/j.gloenvcha.2021.102353
- Jacobi, C., van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, *4*(1), 89–106. https://doi.org/10.1080/21670811.2015.1093271

References

- Maier, D., Niekler, A., Wiedemann, G., & Stoltenberg, D. (2020). How Document Sampling and Vocabulary Pruning Affect the Results of Topic Models. *Computational Communication Research*, *2*(2), 139–152. https://doi.org/10.5117/CCR2020.2.001.MAIE
- Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., Pfetsch, B., Heyer, G., Reber, U., Häussler, T., Schmid-Petri, H., & 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. https://doi.org/10.1080/19312458.2018.1430754
- Nicholls, T., & Culpepper, P. D. (2021). Computational Identification of Media Frames: Strengths, Weaknesses, and Opportunities. *Political Communication*, *38*(1–2), 159–181. https://doi.org/10.1080/10584609.2020.1812777
- Quinn, K. M., Monroe, B. L., Colaresi, M., Crespin, M. H., & Radev, D. R. (2010). How to Analyze Political Attention with Minimal Assumptions and Costs. *American Journal of Political Science*, *54*(1), 209–228. https://doi.org/10.1111/j.1540-5907.2009.00427.x
- Roberts, M. E., Stewart, B. M., & Tingley, D. (2016). Navigating the Local Modes of Big Data: The Case of Topic Models. In R. M. Alvarez (Ed.), *Computational Social Science* (pp. 51–97). Cambridge University Press. https://doi.org/10.1017/CBO9781316257340.004
- Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., Albertson, B., & Rand, D. G. (2014). Structural Topic Models for Open-Ended Survey Responses. *American Journal of Political Science*, *58*(4), 1064–1082. https://doi.org/10.1111/ajps.12103
- Van Atteveldt, W., Trilling, D., & Calderón, C. A. (2022). *Computational Analysis of Communication*. Chapter 11.5 on “Unsupervised Text Analysis: Topic Modeling”. Wiley Blackwell. Link to online chapter 11.5
- Wilkerson, J., & Casas, A. (2017). Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges. *Annual Review of Political Science*, *20*(1), 529–544. https://doi.org/10.1146/annurev-polisci-052615-025542