

Topic Modelling

Implementation des LDA Modells (Gensim) auf ein Korpus von Interviewfragen des NDR Corona Podcast

Inhalt

- Einführung Topic Modelling
- LDA-Modell von Gensim
- Anwendung auf Corona Podcast
- Fazit/Ausblick

1. Topic Modelling

Topic Modelling

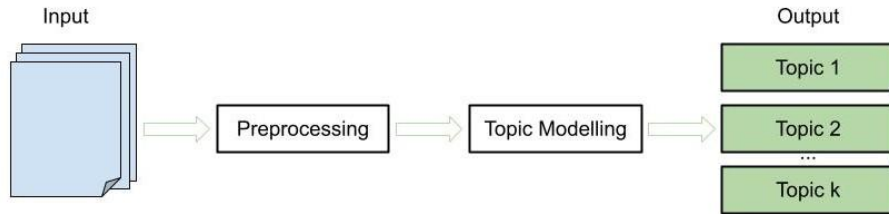


Abb. 1: vereinfachter Ablauf Topic Modelling

- Gebiet des unüberwachten maschinellen Lernens zuzuordnen
- Verfahren zur automatisierten Inhaltsanalyse
- Beschreibung bzw. Zusammenfassung der latenten Struktur großer Datensammlungen
- Berechnung von Themen aus Worthäufigkeiten in Dokumenten
- bekannteste Verfahren: LDA, CTM, STM

Topic Modelling bietet
Methoden zum
automatischen
Organisieren, Verstehen,
Suchen und
Zusammenfassen großer
Datenmengen an.

- Aufdecken latenter
Themen in Sammlungen
- Klassifizierung der
Elemente einer
Sammlung
- Optimierung von
Suchprozessen

Topic Modelling

Annahme:

Ein Korpus besteht aus D Dokumenten (einzelne Dokumente als d_1, d_2, \dots) und V Wörtern bzw. Termen (= alle Wörter, die im gesamten Korpus vorkommen, einzelne Wörter als v_1, v_2, \dots).

Es kommen latente Themen K zu unterschiedlichen Anteilen in den Dokumenten D vor, K wird vor Beginn festgelegt.

Alle Wörter V gehören zu unterschiedlichen Wahrscheinlichkeiten zu den Themen K .

(Vgl. Unkel 2020)

Topic Modelling

Ziel:

Berechnung der Matrix $D \times K$: für jedes Dokument d und jedes Thema k wird die Wahrscheinlichkeit berechnet, dass das Thema im Dokument vorkommt

Berechnung der Matrix $V \times K$: für jedes Wort w und jedes Thema k wird die Wahrscheinlichkeit berechnet, dass das Wort im Thema vorkommt

→ Beschreibung und Interpretation der Themen

$$\begin{array}{l} D \times K \\ \left[\begin{array}{l} k_1: 0.5 \\ k_2: 0.3 \\ k_3: 0.1 \\ k_n: 0.2 \end{array} \right] d_n \\ \\ V \times K \\ \left[\begin{array}{l} v_1: 0.1 \\ v_2: 0.4 \\ v_3: 0.1 \\ v_n: 0.2 \end{array} \right] k_n \end{array}$$

Abb. 2: Beispielmatrixen

2. Latent Dirichlet Allocation (LDA)

LDA

- generatives probabilistisches Modell für Datensammlungen (z.B. Textkorpora)
 - Technik um Dimension von Datensammlungen zu reduzieren
 - Anwendung auf Big Data
- Ziel: Finden von kurzen Beschreibungen zu den Elementen einer Datensammlung zur effizienten Verarbeitung großer Datenmengen
- Unterschied zu einem einfachen Dirichlet-multinomial Clustermodell: Dokumente werden mehrere Themen zugeordnet

(Vgl. Blei et al. 2003)

LDA

Annahmen:

- Jedes Dokument ist eine Sammlung von Wörtern bzw. 'Bag-of-Words'.
 - Reihenfolge von Wörtern + grammatische Rollen werden nicht berücksichtigt
 - "assumption of exchangeability" (Blei et al. 2003: 994)
- Anzahl der Themen k ist im Voraus festgelegt

the red dog

cat eats dog

dog eats food

red cat eats

	the	red	dog	cat	eats	food
the red dog	1	1	1	0	0	0
cat eats dog	0	0	1	1	1	0
dog eats food	0	0	1	0	1	1
red cat eats	0	1	0	1	1	0

Abb. 3: Beispiel Bag-of-Words

LDA

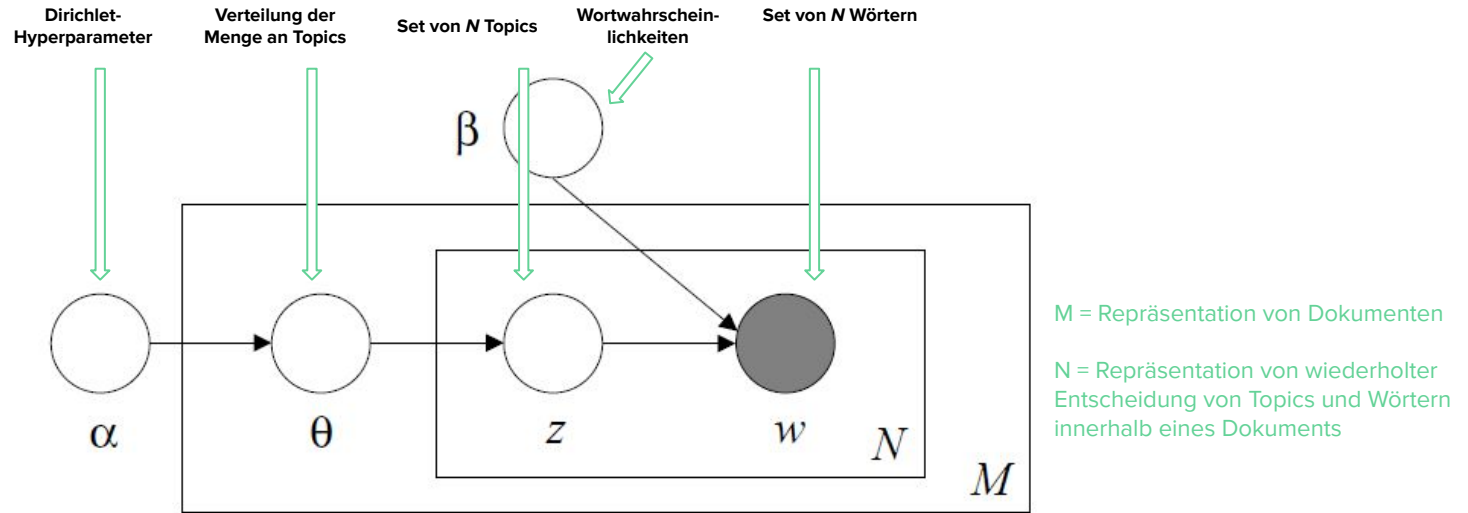


Abb. 4: Graphische Repräsentation des LDA-Modells (Blei et al. 2003: 997)

LDA

Ablauf - Berechnung der Wortwahrscheinlichkeiten ($V \times K$):

1. Jedem Wort w eines Dokuments d wird eins von n Themen K zufällig zugeordnet. n wurde im Voraus festgelegt.
2. Für jedes Dokument d wird für jedes Wort w folgendes berechnet:
 - a. $p(k | d)$: Anteil der Wörter im Dokument d , die dem Thema k zugeordnet sind, das aktuelle Wort wird nicht betrachtet
 - b. $p(w | k)$: Anteil der Zuordnungen zum Thema k an allen Dokumenten, die von diesem Wort w stammen
3. Berechnung der Wahrscheinlichkeit für aktuelles Wort mit zugewiesenen Thema:

$$p(w \text{ mit } k) = p(k | d) * p(w | k)$$

(Vgl. Kulshrestha 2019)

3. LDA-Modell von Gensim

Was ist Gensim?



URL:
https://radimrehurek.com/gensim/_images/gensim_logo_positive_complete_tb.png

- frei verfügbare Open-Source Python-Bibliothek
- aufrufbar unter:
<https://radimrehurek.com/gensim/>
- Bereitstellung unüberwachter maschineller Lernalgorithmen zur Verarbeitung unstrukturierter digitaler Texte
 - Darstellung dieser als semantische Vektoren
 - benötigt keine manuelle Annotation, nur ein “plain-text”-Korpus
 - z.B. Word2Vec, FastText, LSI, LDA

Gensim

Kernkonzepte:

- Dokumente
- Korpora
- Vektoren
- Modelle

LDA-Modell als eine
Umwandlung von
Bag-of-Words-Zählungen in
einen Themenraum mit
geringerer Dimensionalität

- Themen von LDA als Wahrscheinlichkeitsverteilungen über Wörter
- Ableitungen der Verteilungen aus dem Trainingskorpus
- Dokumente als Mischung dieser Themen

LDA-Modell von Gensim

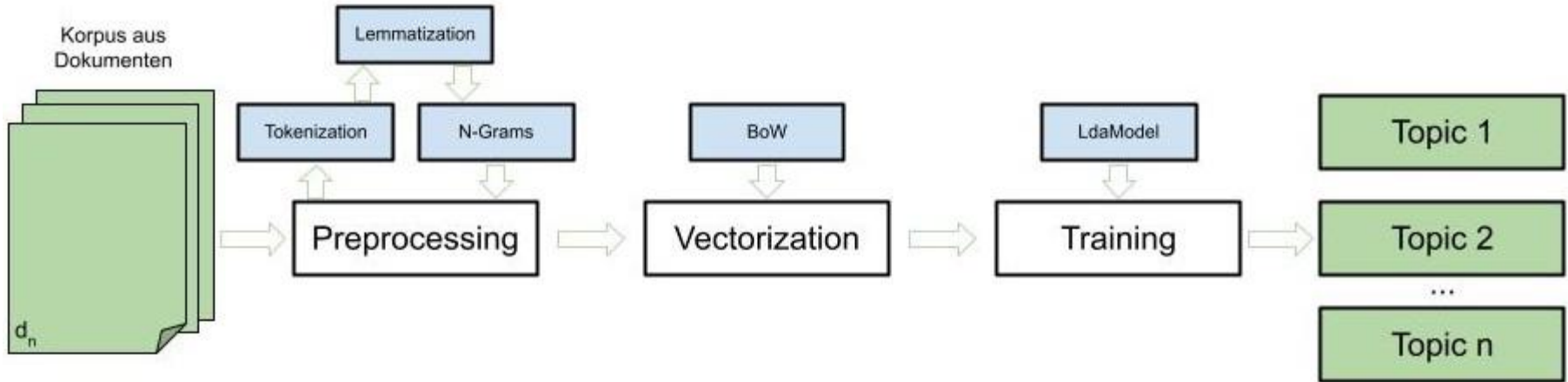


Abb. 5: Ablauf LDA Gensim

4. Implementation LDA Corona Podcasts

Implementation LDA-Modell auf Corona Podcasts

- Implementation in Python 3.9
- Code verfügbar unter:
https://github.com/ChristineSchaefer/topicmodel_coronapodcast.git
- Korpus: an die Wissenschaftlerin Sandra Ciesek und an den Wissenschaftler Christian Drostén gestellte **Fragen** im NDR Corona Podcast (Folge 1 - Folge 68)
 - Fragen extrahiert aus verfügbaren PDFs und abgespeichert in csv-Datei
- verwendete Bibliotheken: SpaCy, NLTK, Gensim
 - Training des LDA-Modells mit Lemmata (Unigram) anstelle von N-Grammen

Erwartungen

1. Mithilfe des LDA-Modells von Gensim können die Fragen, die in den Podcasts gestellt werden, inhaltlich zusammengefasst werden. Entsprechende Topics können benannt werden.
2. Aufgrund der geringen Anzahl an zugrunde liegenden Fragen bzw. aussagekräftigen Wörtern innerhalb dieser sind ermittelte Topics nicht repräsentativ für die Themen innerhalb einer Folge.

Datengrundlage

	Sandra Ciesek	Christian Drost
Anzahl der Interviews	8	57
Fragen/Dokumente	285	1135
Unique Tokens (insg.)	288	810
Unique Tokens (filter)	17	67
Fragen/Dokumente (filter)	219	837

Tab. 1: Anzahl der verfügbaren Interviews, Fragen und Tokens als Input für das LDA-Modell

Implementation

Trainingsparameter:

- *num_topics*: Anzahl der Themen abhängig von Daten und Anwendung
 - relativ kleines Korpus → Anzahl der Themen gering
- *chunksize*: Anzahl der Dokumente, die gleichzeitig verarbeitet werden sollen
- *passes*: Anzahl des Trainings auf das gesamte Korpus
- *iterations*: Anzahl der Wiederholung einer Schleife über jedes Dokument

```
# Set training parameters.
num_topics = 5      # number of topics
chunksize = 2000    # number of processed documents at a time
passes = 20         # how often the model will be trained
iterations = 400    # how often a particular loop will be repeated over each document
eval_every = 1      # evaluate each iteration

# make a index to word dictionary.
temp = dictionary[0] # "load" the dictionary.
id2word = dictionary.id2token

# set model parameters
# Initialisierung (Training) des Transformationsmodells (LdaModel)
model = LdaModel(
    corpus=corpus,
    id2word=id2word,
    chunksize=chunksize,
    alpha='auto',
    eta='auto',
    iterations=iterations,
    num_topics=num_topics,
    passes=passes,
    eval_every=eval_every
)
```

Abb. 6: Code-Ausschnitt Training LDA

Evaluation

- Vergleich der beiden Modelle durch Berechnung der *Perplexity*:

$$\text{perplexity}(D_{\text{test}}) = \exp \left\{ -\frac{\sum_{d=1}^M \log p(\mathbf{w}_d)}{\sum_{d=1}^M N_d} \right\}.$$

Abb. 7: Formale Darstellung der Perplexity (Brim et al. 2003: 1008)

- Evaluation nach jeder Runde des Trainings, nach 20 Runden:

	Sandra Ciesek	Christian Drosten
Perplexity	84.7	139.0
Perplexity pro Wort	-6.405	-7.119
Topic Difference	0.000707	0.002623

Tab. 2: Evaluationsergebnisse LDA ohne gefilterte UT

Sind die ermittelten Themen
verständlich?

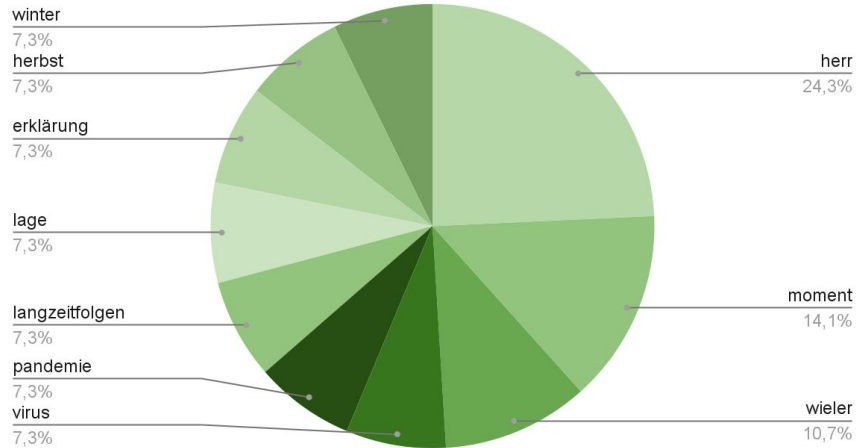
Sind die Themen **kohärent?**

Dient das Themenmodell dem
Zweck, für den es verwendet
wird?

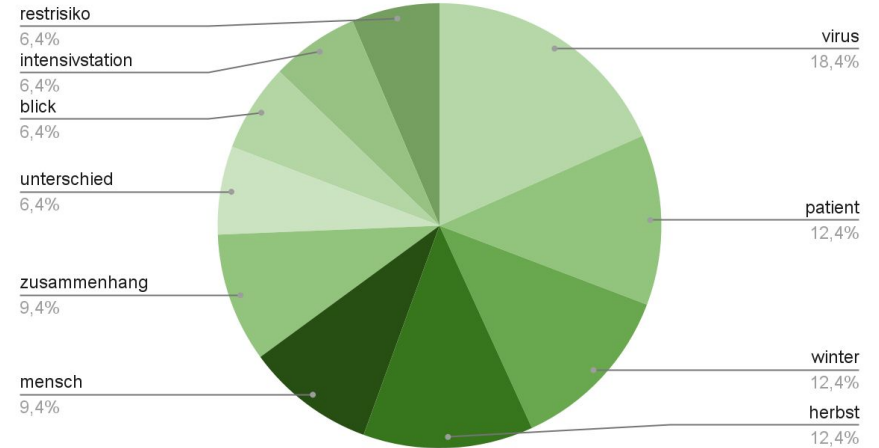
Ergebnisse

Topics Sandra Ciesek

Topic 1



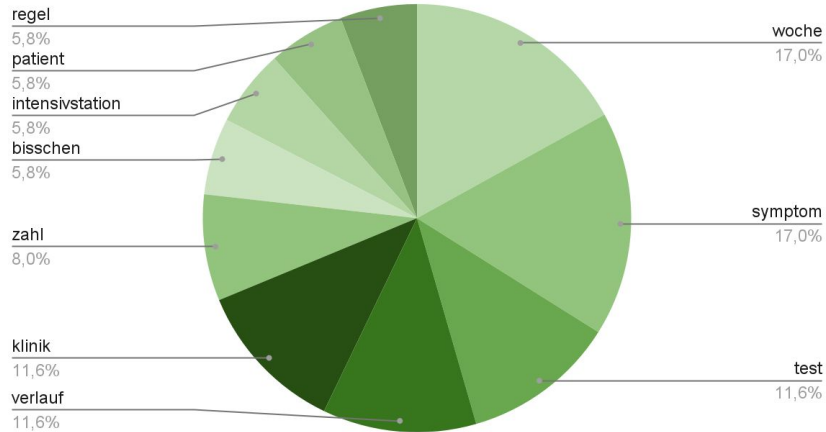
Topic 2



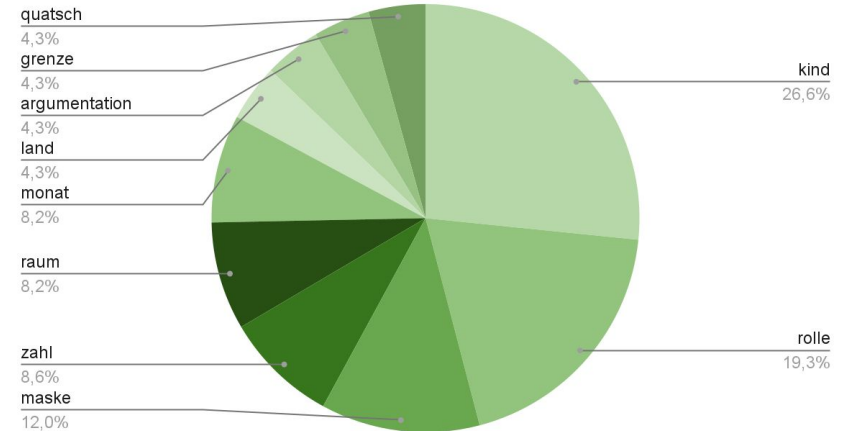
Ergebnisse

Topics Sandra Ciesek

Topic 3



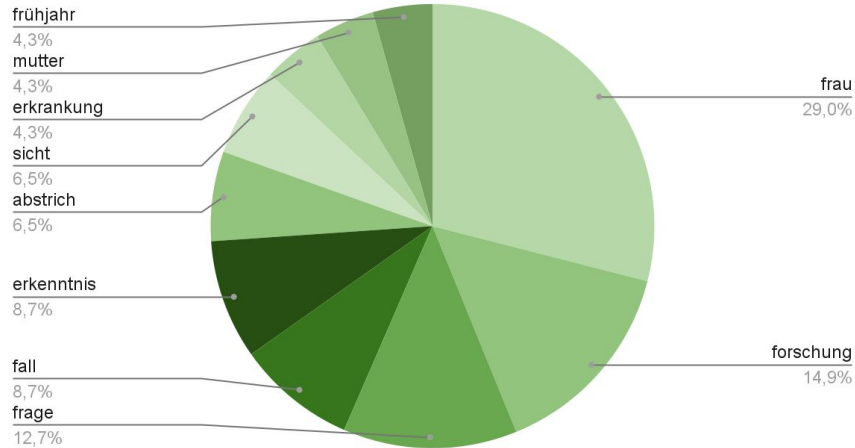
Topic 4



Ergebnisse

Topics Sandra Ciesek

Topic 5



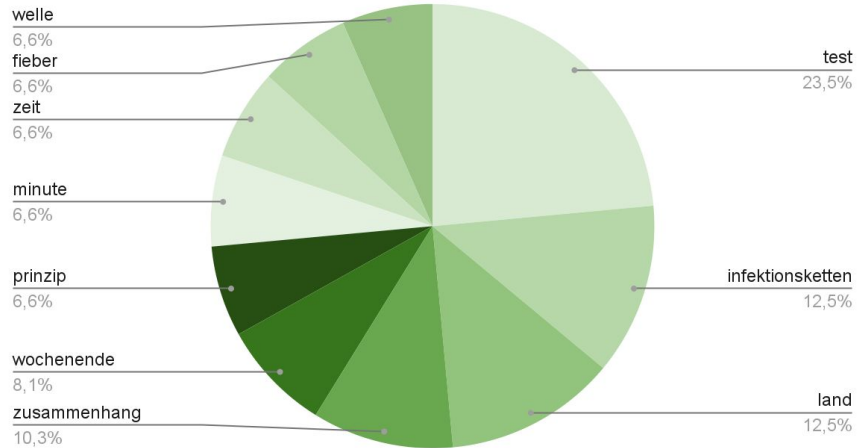
Topic 1	0.121
Topic 2	0.116
Topic 3	0.135
Topic 4	0.110
Topic 5	0.160

Tab. 3: Wahrscheinlichkeiten der Themen SC

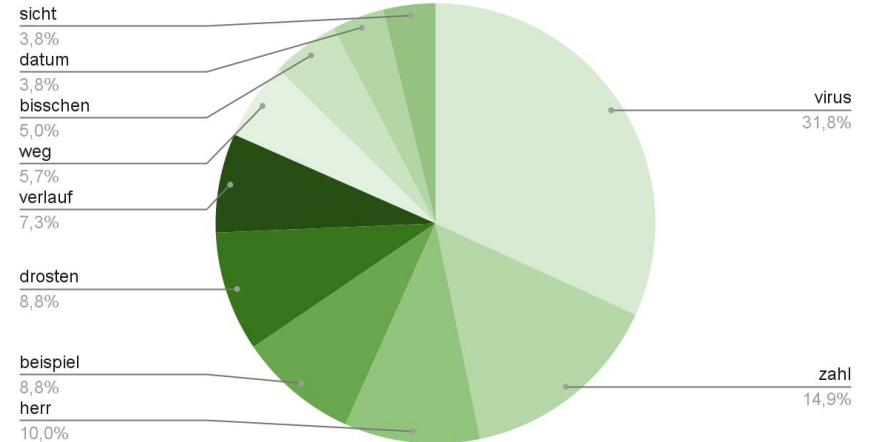
Ergebnisse

Topics Christian Drosten

Topic 1



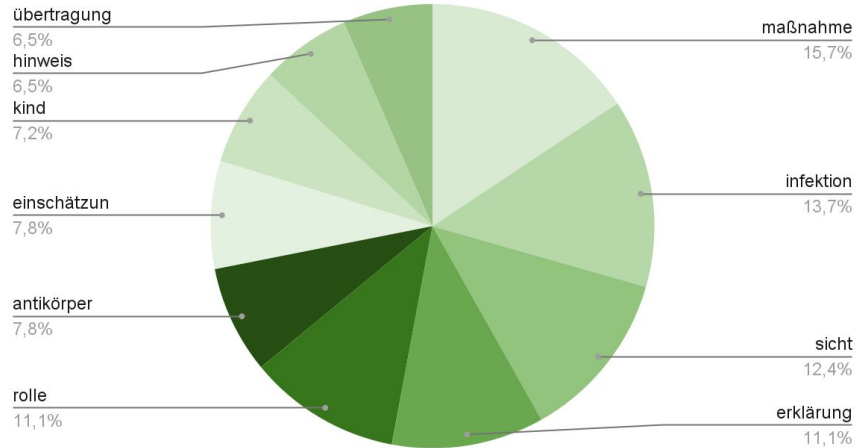
Topic 2



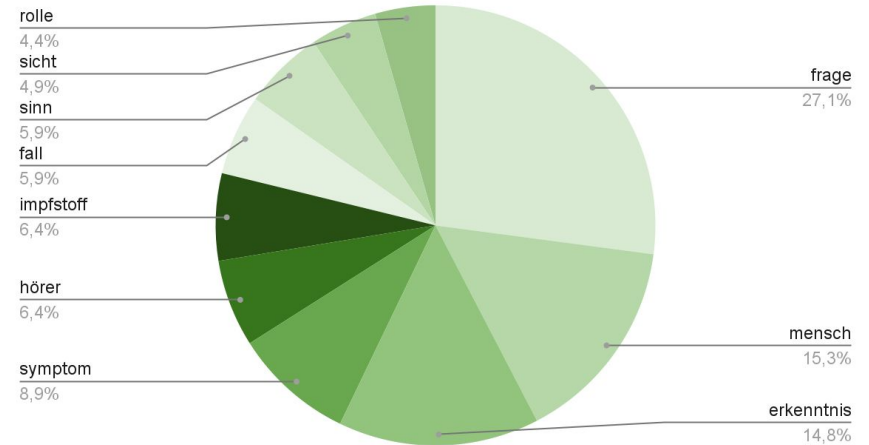
Ergebnisse

Topics Christian Drosten

Topic 3



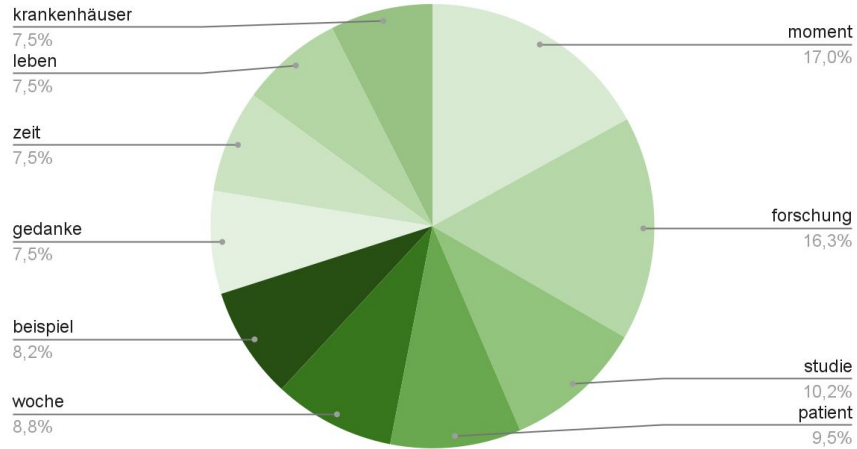
Topic 4



Ergebnisse

Topics Christian Drost

Topic 5



Topic 1	0.105
Topic 2	0.189
Topic 3	0.118
Topic 4	0.160
Topic 5	0.130

Tab. 4: Wahrscheinlichkeiten der Themen CD

5. Fazit und Ausblick

Fazit und Ausblick

- Anwendung des LDA Modells von Gensim auf ein Korpus von Fragen des NDR Corona Podcast als gute Möglichkeit, latente Themen innerhalb dieser sichtbar zu machen
 - weniger Arbeit als manuelle Einsicht
 - Interpretation der Themen (und deren Zusammenhänge) als möglich erachtet
 - Aber:
 - zu kleines Korpus
 - Einbezug von Wörtern, die nicht bedeutungstragend für das Thema sind, aber trotzdem gewichtet wurden (z.B. Frau oder Herr)
- stärkerer Filter für bessere Interpretierbarkeit
- je größer das Korpus, desto besser die Performance
- Einbezug der restlichen Folge um mehr Aufschluss über tatsächliche Themen zu bekommen

Fragen?

Literatur

Blei, David M. et al. (2003): *Latent dirichlet allocation*. J. Mach. Learn. Res. 3, null (3/1/2003). S. 993–1022.

Hoffman, Matthew D. et al. (2010): *Online Learning for Latent Dirichlet Allocation*. Advances in Neural Information Processing Systems. 23. S. 856-864.

Kulshrestha, Ria (2019): *A Beginner's Guide to Latent Dirichlet Allocation(LDA)*. A statistical model for discovering the abstract topics aka topic modeling. Abrufbar unter: <https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2> (zuletzt aufgerufen: 18.10.2021)

Unkel, Julian (2020): *Topic Modeling*. In: Computational Methods in der politischen Kommunikationsforschung. Methodische Vertiefung: Computational Methods mit R und RStudio. Abrufbar unter: <https://bookdown.org/joone/ComputationalMethods/topicmodeling.html> (zuletzt aufgerufen: 18.10.2021)