

# Topic Modeling mit deutschen Tweets

Julia Karst, Tobias Niedballa, Sarah Ondraszek, Julia Weyer

Computerlinguistische Programmierung

Sommersemester 2021

14.07.2021

# Gliederung

- Ziel
- Tweets sammeln
- Preprocessing
- LDA Model
- Visualisierung
- GUI
- Interpretation der Topics
- Fazit

# Ziel

Es sollten Topics für ein Tweetskorpus der ersten Maiwoche mit Hilfe von LDA Gensim generiert werden. Die Skripte und Daten finden sich unter [https://github.com/sarahondraszek/comp\\_ling\\_LDA](https://github.com/sarahondraszek/comp_ling_LDA).

Unsere Hypothese: Die Twitter-NutzerInnen haben hauptsächlich über Covid-19 und aktuelle politische Ereignisse geschrieben.

# Tweets sammeln

- Beantragen eines Twitter Developer Accounts für Studierende
- Zugriff auf die Twitter API (mit Bearer-Token)
- Nutzung der bereits vordefinierten Zugriffsmethoden
- Implementierung in Python
  - ▶ Erste Version mit requests, json
  - ▶ Zweite und finale Version mit 'searchtweets' und entsprechender Query

```
# request params for this query
query = searchtweets.gen_request_parameters("a lang:de", start_time="2021-05-08T00:00",
                                           end_time="2021-05-08T23:59",
                                           results_per_call=100)
```

Beispiel-Query für API-Zugriff

# Tweets sammeln

- 'ResultStream' Output mit 10.000 Tweets für jeden Tag des Zeitraums 04.05.2021 bis 10.05.2021
- Gesammelte Daten werden gespeichert als CSV-Dateien

# Preprocessing

- Text aus .csv filtern
- mit Hilfe von regulären Ausdrücken überflüssige Zeichen entfernen
  - ▶ Nutzernamen
  - ▶ Hashtags
  - ▶ Links
  - ▶ twitterspezifische Marker (RT, NAN, via)
- Tokenisierung
- Entfernung der Stoppwörter und aller Token der Länge 2 oder weniger
- Lemmatisierung
- N-Gramme
- Sprachfilter
- Bag-Of-Words

# LDA

- Verwendung von Gensim LDA
- Laden der einzelnen vorprozessierten Dateien als Input für das Modell
- Trainings-Parameter angepasst an unser Korpus
- Modell wird zwischengespeichert (für Visualisierung)

# Visualisierung

- Um die Topics zu visualisieren wird pyLDAvis genutzt
- Probleme mit der neuesten Version → besser: pyLDAvis 2.1.2
- Weitere Alternative: Ergebnisse als HTML speichern und dann ansehen
- neueste Version: *import pyLDAvis.gensim\_models*  
Version 2.1.2: *import pyLDAvis.gensim*

```
18 lda_visualization = pyLDAvis.gensim.prepare(model, corpus, tweet_dictionary)
19 pyLDAvis.show(lda_visualization)
```

Visualisierung mit pyLDAvis 2.1.2



# GUI

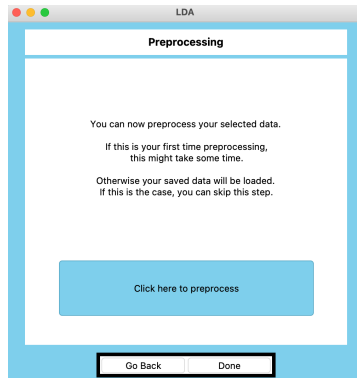
Die GUI wurde mit tkinter erstellt. Um Darstellungsprobleme unter macOS zu beheben, wird zusätzlich tkmacosx für die Buttons benutzt.

Insgesamt gibt es sechs Seiten:

- 1 Startbildschirm
- 2 Korpusauswahl
- 3 Preprocessing
- 4 Modellparameter
- 5 Visualisierung
- 6 Endbildschirm

# GUI

- blauer Hintergrund
- drei Frames
  - ▶ oben: Titel
  - ▶ mittig: Inhalt
  - ▶ unten: Seiten wechseln
- Platzierung mit *place*:  
relative Angaben zu Koordinaten,  
Breite und Höhe



# GUI: Korpusauswahl und Preprocessing

- können nach erstmaliger Nutzung übersprungen werden
- Button „Select Files“ ermöglicht die Auswahl mehrerer csv-Dateien
- Preprocessing wird auf den ausgewählten Dateien durchgeführt
- das Ergebnis wird zwischengespeichert

# GUI: Modellparameter und Visualisierung

- Parameter:  
no\_below, no\_above, number of topics, chunk size, passes, iterations
- Default-Werte als Hilfestellung
- Eingaben werden überprüft
- Topics werden zunächst auf der Konsole ausgegeben
- die Visualisierung mit pyLDAvis kann anschließend im Browser geöffnet werden

# Interpretation der Ergebnisse

Häufig vorkommende Themen:

- Corona und Impfstoffe
- Auswirkungen der Pandemiepolitik  
→ Reaktion auf Kolumne von Sascha Lobo



# Interpretation der Ergebnisse

- René Benko

→ Reaktion auf Beitrag des ZDF Magazin Royale



- Auschwitz Memorial
- Demonstrationen am Tag der Arbeit

# Interpretation der Ergebnisse

Probleme bei der Interpretation:

- Language Identification nicht akkurat genug
- Korpus dominiert von Retweets
- Typische Topic Modeling Probleme

# Fazit

- Topics analysiert ✓
- Hypothese bekräftigt ✓  
⇒ pandemische & politische Lage stehen im Fokus
- leicht zu interpretierende Topics ✗
- Verbesserungen: genauere Sprachfilter, anderes TM Modell/Verfahren