Part III: Feature Extraction & Machine Learning

Part III: Feature Extraction & Machine Learning

Theoretical Background

Outline

i. Word Embeddings
ii. Feature Extraction
iii. Background Machine Learning
    i. Training
    ii. Evaluation
    iii. Implementation in R
iv. Visualization in R

Feature Extraction

- Preprocess input for ML or DL algorithms / classification tasks: convert the text data into a numeric, structured format
- Goal: Mapping from vocabulary to real number vectors → Word Embeddings
- Methods: Vocabulary based, Neural networks, Co-occurrence matrix (GloVE), etc.

Vocabulary based Feature Extraction

- Bag-Of-Words (BOW)
  - Create a vocabulary with all occurring words in documents
  - Assumption: each word is independent from the others that are present in the document
  - No examination of word order
  - Each document is represented by the **term frequency vector** (occurrence of all the distinct words that are present in the document)
  - Term Frequency - Inverse Document Frequency:
    → Does not imply that all terms are considered equally important
    → Idea: Penalize words that are too frequent

Example (BOW)

Documents

"Die Ausgrenzung von MigrantInnen ist inakzeptabel und rassistisch."

"Die Maskenpflicht ist sinnvoll."

"Die Diskriminierung von Frauen ist inakzeptabel."

Vector-space representations

	die	ausgrenzung	von	migrantinnen	ist	inakzeptabel	und	rassistisch	maskenpflicht	sinnvoll	diskriminierung	frauen
Doc1	1	1	1	1	1	1	1	1	0	0	0	0
Doc2	1	0	0	0	1	0	0	0	1	1	0	0
Doc3	1	0	1	0	1	1	0	0	0	0	1	1

Example (TF-IDF)

Documents

"Die Ausgrenzung von MigrantInnen ist inakzeptabel und rassistisch."

"Die Maskenpflicht ist sinnvoll."

"Die Diskriminierung von Frauen ist inakzeptabel."

Vector-space representations

	die	ausgrenzung	von	migrantinnen	ist	inakzeptabel	und	rassistisch	maskenpflicht	sinnvoll	diskriminierung	frauen
Doc1	0	0.48	0.18	0.48	0	0.18	0.48	0.48	0	0	0	0
Doc2	0	0	0	0	0	0	0	0	0.48	0.48	0	0
Doc3	0	0	0.18	0	0	0.18	0	0	0	0	0.48	0.48

Vocabulary based Feature Extraction

- Term frequency - inverse document frequency (BOW) – Example:
  - Dokument 1: he likes eating banana
  - Dokument 2: she likes eating cakes he likes drinking banana juice
  - Dokument 3: he likes drinking tomato juice

Dokument	he	likes	eating	banana	she	cakes	drinking	juice	tomato
1	1	1	1	1	0	0	0	0	0
2	1	2	1	1	1	1	1	1	0
3	1	1	0	0	0	0	1	1	1

Unsupervised Word Vectors

- GloVe
  - Idea: Model the semantic importance of a word in a numeric form
  - Make use of the co-occurrence matrix
  - Learned representations for words: same meaning = similar representation in the vector space
  - Enable performing mathematical operations on it:
    - ϕ: $W \rightarrow R^n$
    - $\phi(\text{"}king\text{"}) - \phi(\text{"}man\text{"}) + \phi(\text{"}woman\text{"}) = \phi(\text{"}queen\text{"})$

Comparison of two approaches

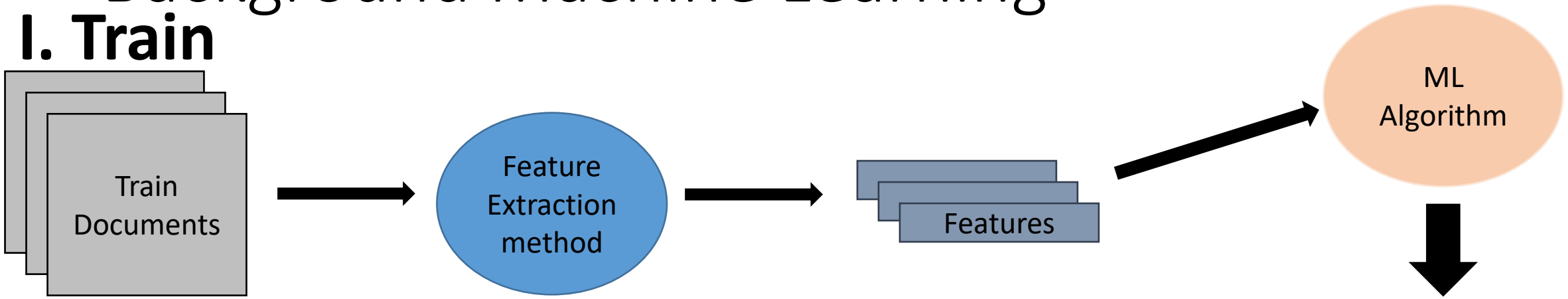
- Both result in vector represenation for each word contained in a corpus
- BoW model does not consider the ordering, semantics
- Lower dimensionality for GloVE (i.e. 100-300), whereas for BOW (i.e. 100000)
- Approximate consideration of word semantics → large and “good” corpus for meaningfull vector representations needed
- Pretrained models usually fail on target tasks with different domain (null vector for unknown, special context words)
- Pretrained models Require large amounts of memory and computational resources

Sentiment Classification

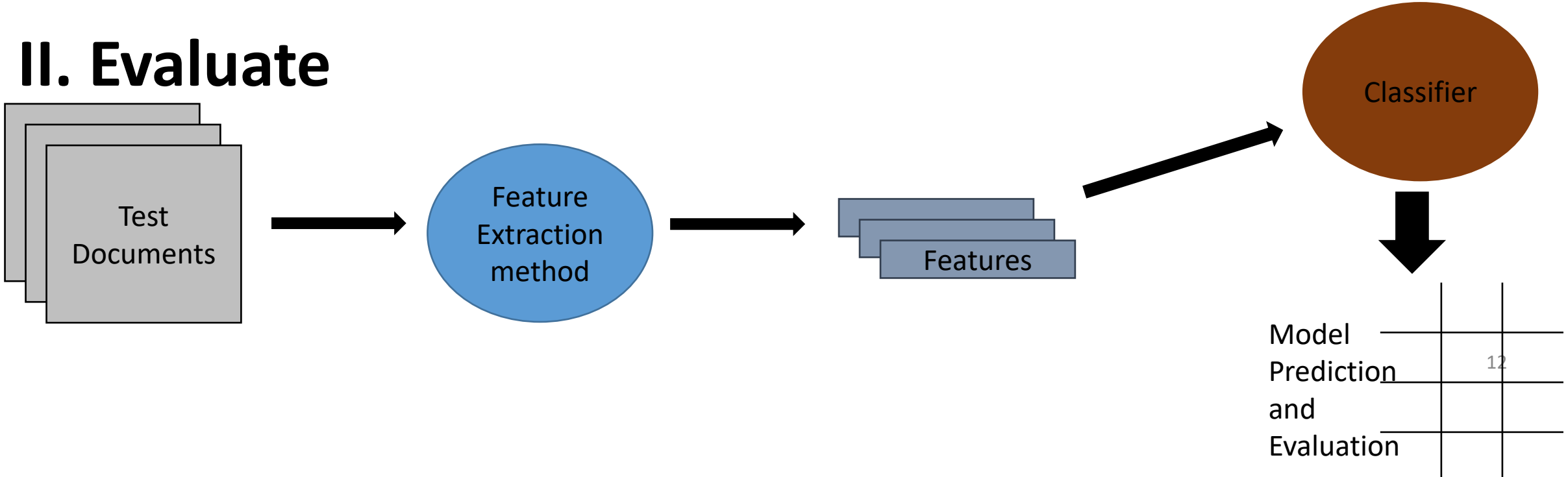
- Classification as supervised machine learning task
- Goal: Make predictions by training the model on annotated data
- Given documents (features) and fixed amount of classes and classification algorithm(s) → learn a classifier → should predict best possible class for each document
- We consider: Binary classification (2 Sentiments as output)
- Our approach in ABSA: Create topic-specific word embeddings

Background Machine Learning

I. Train



II. Evaluate



Background Machine Learning

- Possible classification algorithm:
  - Logistic regression with regularization
  - Naive Bayes
  - Random Forests
  - Support Vector Machine

- Train-Test-Split (Cross Validation) for a reliabel assessment
- Evaluate the prediction goodness with metrics: Accuracy, Recall, Precision, F1-Score

- Model evaluation via Confusion Matrix:

	Predicted: NO	**Predicted: YES**	
Actual: NO	**T**rue **N**egative	**F**alse **P**ositive	
Actual: YES	**F**alse **N**egative	**T**rue **P**ositive	

- Evaluation metrics:
  - Accuracy: $$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
  - F1-Score: $$F_1 = 2 * \frac{precision * recall}{precision + recall} \qquad precession = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

Visualization in R

Packages we recommend:

- **Quanteda**: preprocessing, corpus construction, tokenization, document-feature matrix, wordscore, topic modelling, visualizations
  - Useful tutorial for the package: https://tutorials.quanteda.io/
- **Tidytext**: preprocessing, feature extraction, visualizations; works well with other tools in wide use (dplyr, tidyr, wordcloud, ggplot2)
  - Useful tutorial for the package: https://www.tidytextmining.com/

Visualization in R

Packages we recommend:

- **Stringr**: easy working with strings, especially regarding regular expressions, pattern matching functions, character manipulation, whitespace management
  - Useful tutorial for the package: https://cran.r-project.org/web/packages/stringr/vignettes/stringr.html
- **Text2vec**: construct dtm, tcm, word embeddings (i.e. GloVE), topic modelling (i.e. LDA)
  - Useful tutorial for the package: http://text2vec.org/index.html

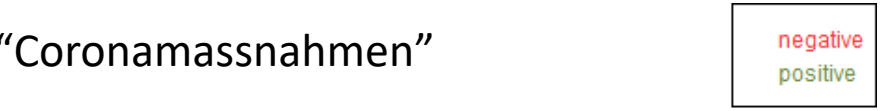
Visualization options

- Wordclouds
- Barplots
- Word frequency plots
- Lexical dispersion plots
- Comparing word usage plots

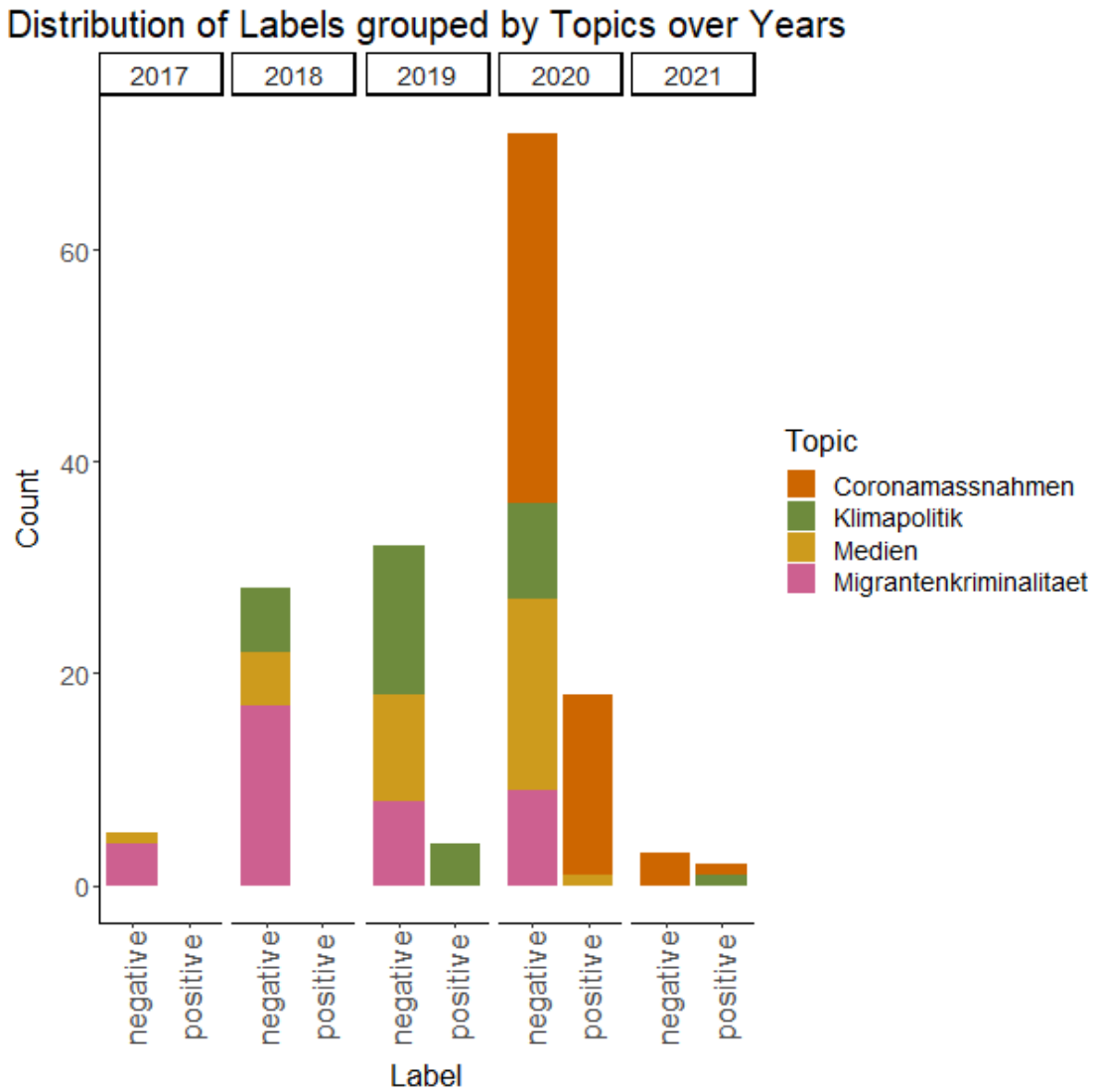
Visualization examples: Wordclouds

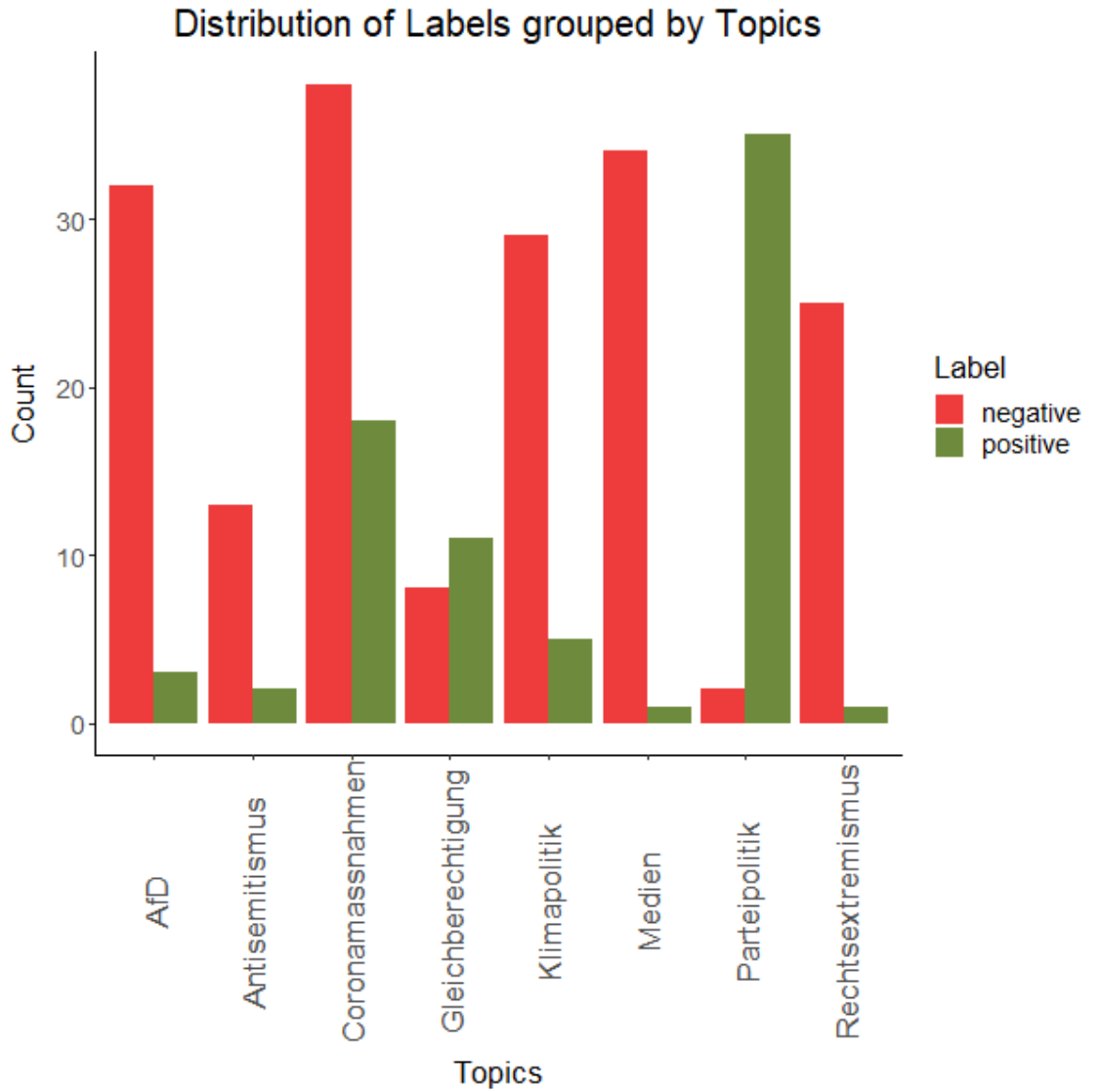
"AfD"

"Coronamassnahmen"

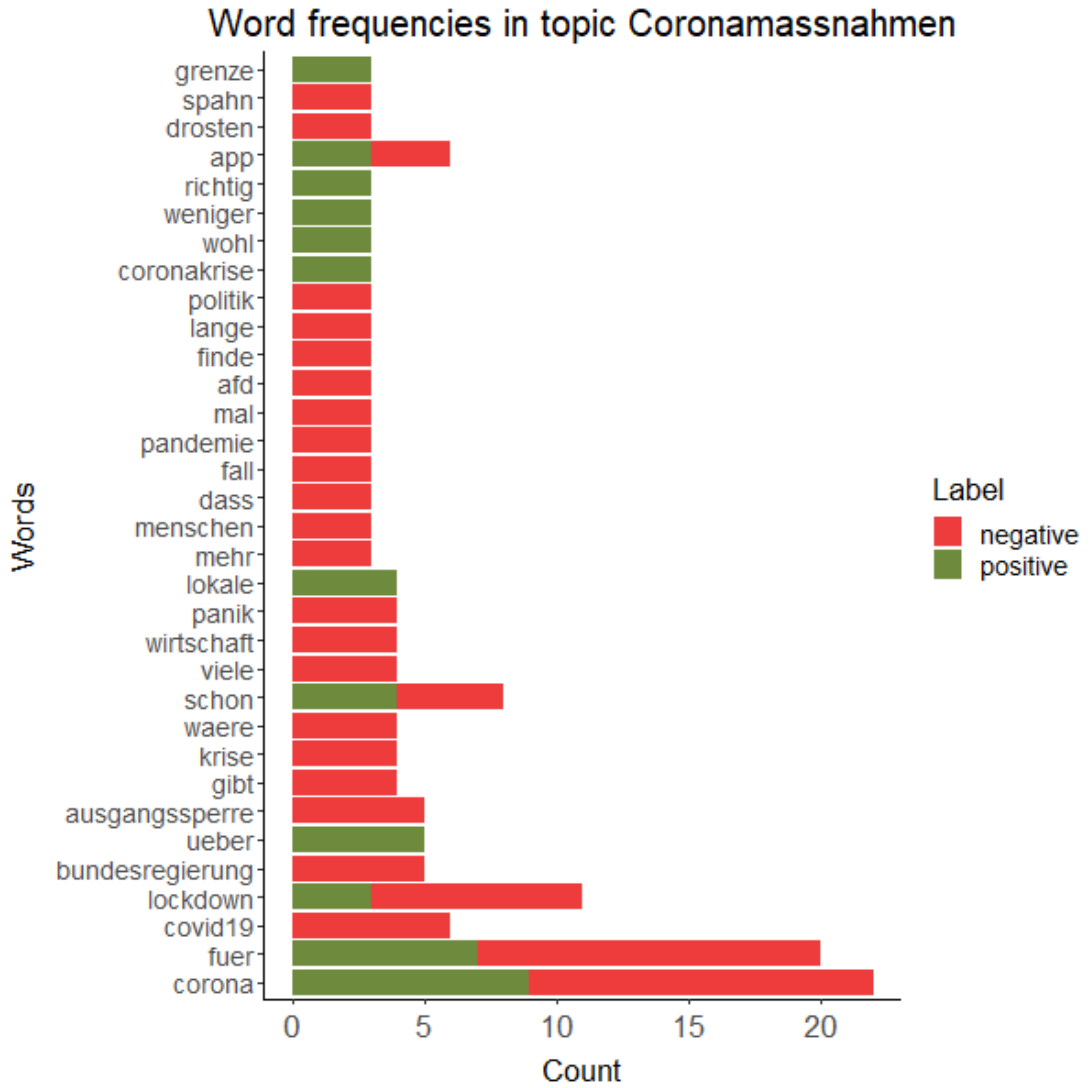


Visualization examples: Barplots





Visualization examples: Word frequencies



Part III: Feature Extraction & Machine Learning

Practical Application

Preparation:

1) Get an access to Google Collab
2) Open Jupyter Notebook: https://colab.research.google.com/drive/1l4ZhBsPsXTfWr8nwvuFHREpHeHgU0Po_#scrollTo=ZM0up_dlsg5e
3) Set Up R

Exercises:

1) Create numerical features from textual data:
    1) Bow (?)
    2) Tf-Idf (?)
2) Visualize different topics:
    1) Print all existing Topics in the data set.
    2) Add 2 other topics to the barplot “Distribution of Labels grouped by Topics over Years”. Choose 2 more colors.
    3) Generate the Wordclouds and the word frequency plot for 1 other topic.

Part III: Feature Extraction & Machine Learning

Literature and References

- Sharafi, A., Wolf, P. and Krcmar, H. (2010). Knowledge discovery in databases on the example of engineering change management., ICDM 2010.
- Sarkar, D. (2016). Text analytics with python.
- Pennington, J., Socher, R. and Manning, C. (2014). Glove: Global vectors for word representation.
- Pennington, Jeffrey and Socher, Richard and Manning, Christopher (2014). Pre-trained word vectors. https://nlp.stanford.edu/projects/glove, abgerufen am 10. Mai 2019.
- Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R. and Delen, D. (2012). Practical text mining and statistical analysis for non-structured text data applications, Academic Press.