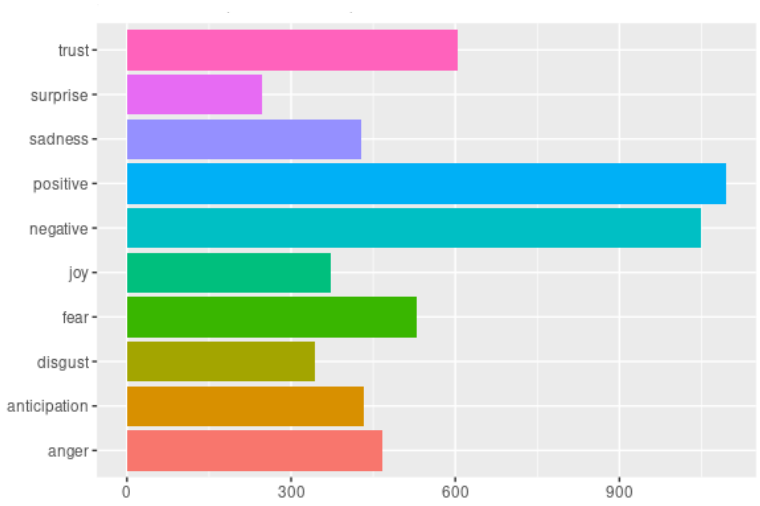
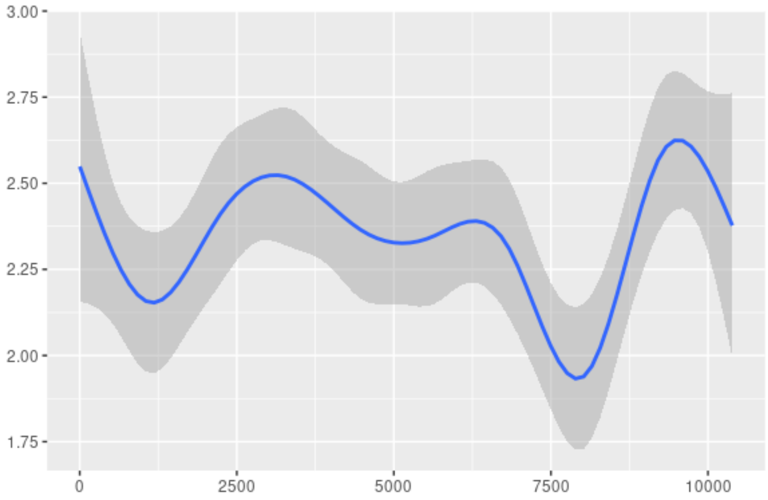
Sentiment Analysis:
Fallstricke eines scheinbar einfachen Tools





#NTF22 Workshop

Prof. Dr. Dennis Klinkhammer

Agenda

- Grundlegende Funktionsweise
- arXiv: Do-It-Yourself-Tutorials
- Fallstricke der Sentiment Analysis
- Referenzen

Grundlegende Funktionsweise

Grundlegende Funktionsweise

- Sentiment Analysis ist ein Untergebiet des Text Mining
- Methodische Grundlagen:
  - Statistik
  - Machine Learning
  - Natural Language Processing -> Deep Learning

[2], [5], [6], [7]

Grundlegende Funktionsweise

- Algorithmenbasierte Verarbeitung von Text- und Sprachdaten
- Übergeordnetes Ziel der Sentiment Analysis

Einschätzung der Haltung

(in Text- und Sprachdaten)

als positiv oder negativ

[5], [6], [7]

Grundlegende Funktionsweise

- Wörtern wird nicht einfach ein Sentiment zugewiesen
- Wörter werden in Vektoren überführt, um deren Bedeutung in einem Satz abbilden zu können
- Gängige Tools, bspw. für Python
  - **Word2Vec** : Word to Vectors
  - **GloVe** : Global Vectors for Word Represantation

[2], [6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism
  - Fascism

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3)

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]
    [0, 0, 1, 0, 2, 1]
    [0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]

    [0, 0, 1, 0, 2, 1]

    [0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]
    [0, 0, 1, 0, 2, 1]
    [0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]

    [0, 0, 1, 0, 2, 1]

    [0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]
  - Position + Häufigkeit (n) [0, 0, 1, 0, 2, 1]

[0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]
  - Position + Häufigkeit (n) [0, 0, 1, 0, 2, 1]

[0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]
  - Position + Häufigkeit (n) [0, 0, 1, 0, 2, 1]

    [0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]
- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1]
  - Position + Häufigkeit (n) [0, 0, 1, 0, 2, 1]

  [0, 0, 0, 1, 3, 1]

[6]

Grundlegende Funktionsweise

- One Hot Encoding und Word Embedding ermöglichen Vektorisierung
- Beispiel für One Hot Encoding:
  - Marxism = [1, 0, 0, 0]
  - Fascism = [0, 1, 0, 0]

Sentiment

⬇

- Beispiel für Word Embedding:
  - Marxism (1) is (2) good (3) -> [1, 0, 0, 0, 1, 1] + [**?**]
  - Position + Häufigkeit (n) [0, 0, 1, 0, 2, 1] + [**?**]

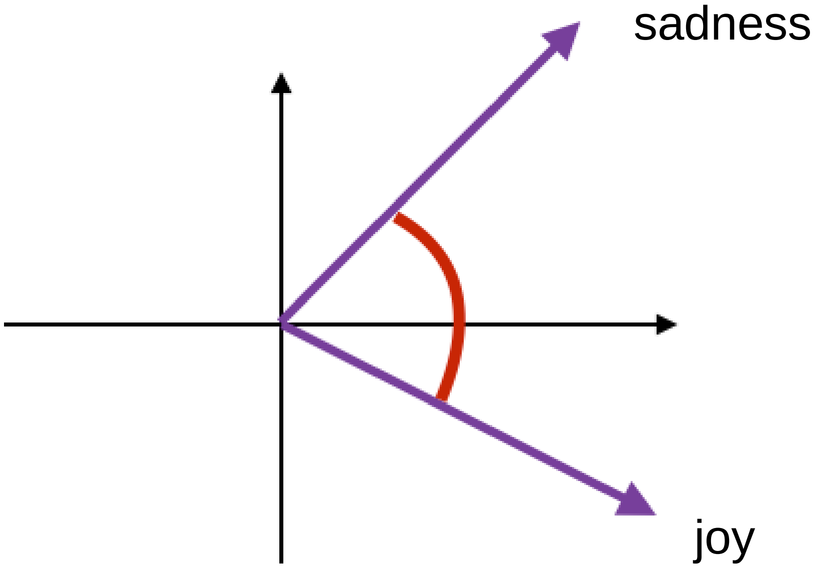
[0, 0, 0, 1, 3, 1] + [**?**]

[6]

Grundlegende Funktionsweise

- **NRC** Word-Emotion Association Lexicon
- 2 Sentiments und 8 Emotions

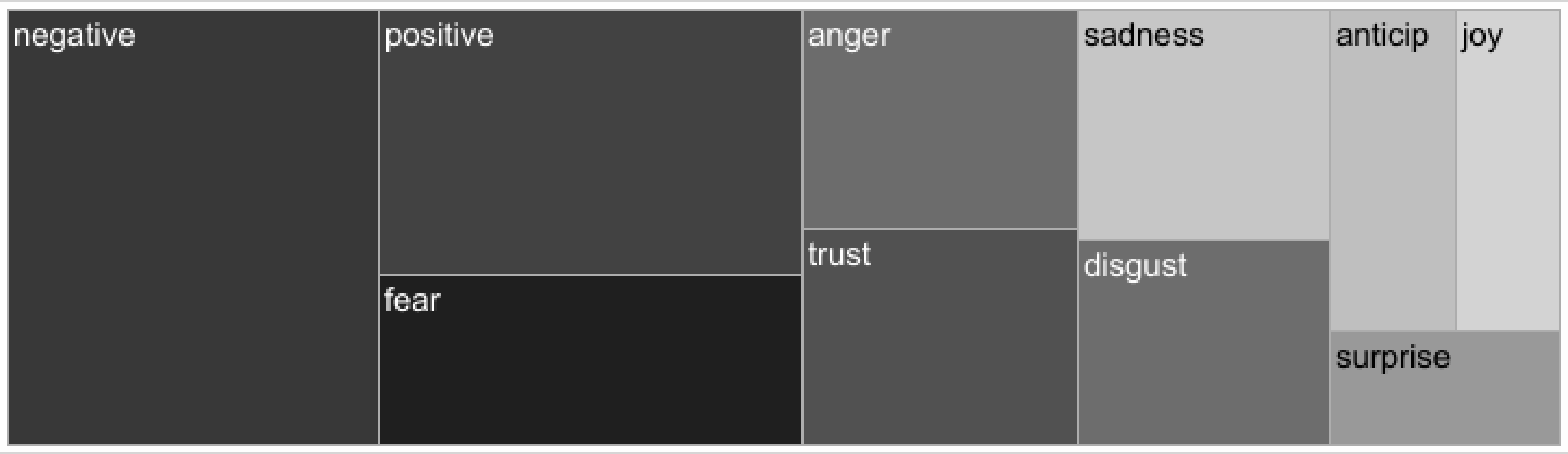
positive	**negative**
trust	fear
anticipation	anger
surprise	sadness
joy	disgust



[6], [7]

Grundlegende Funktionsweise

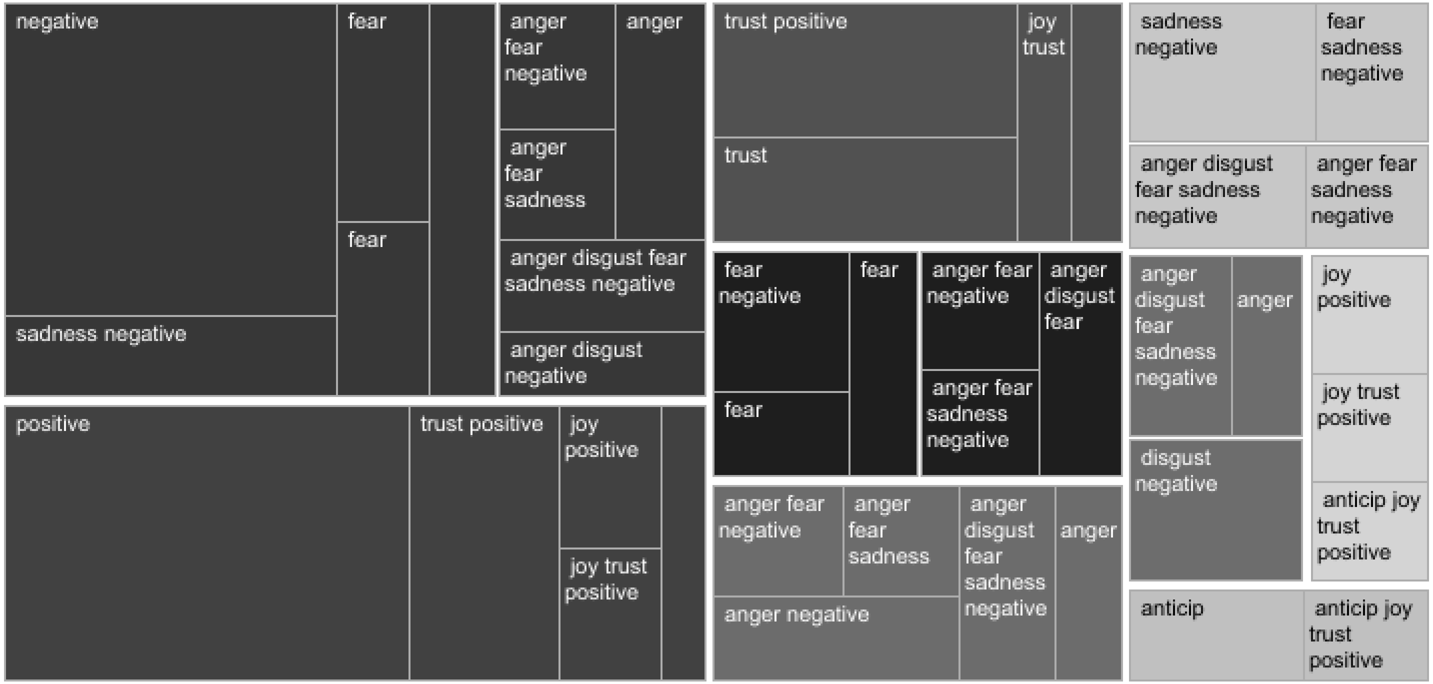
- NRC Word-Emotion Association Lexicon **Affect Categories**



[7]

Grundlegende Funktionsweise

- NRC Word-Emotion Association Lexicon **Set of Categories**



[7]

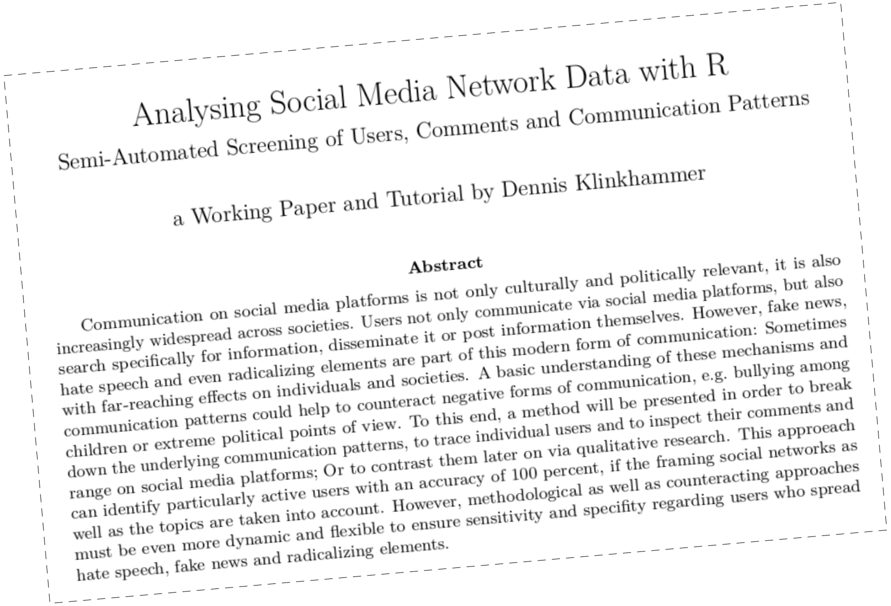
arXiv: Do-It-Yourself-Tutorials

arXiv: Do-It-Yourself-Tutorials

- Aktuelle Entwicklungen der Sentiment Analysis lassen sich zum gegenwärtigen Zeitpunkt am besten auf arXiv nachvollziehen

arXiv: Do-It-Yourself-Tutorials

- Aktuelle Entwicklungen der Sentiment Analysis lassen sich zum gegenwärtigen Zeitpunkt am besten auf arXiv nachvollziehen



Analysing Social Media Network Data with R

Semi-Automated Screening of Users, Comments and Communication Patterns

a Working Paper and Tutorial by Dennis Klinkhammer

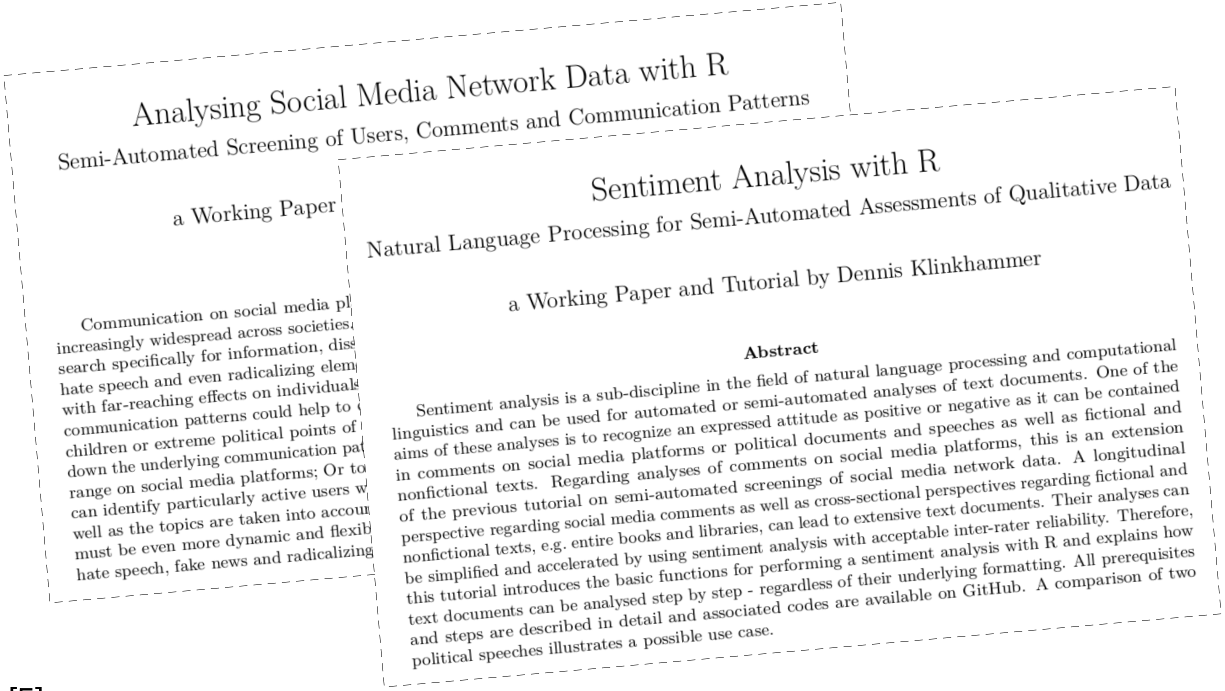
Abstract

Communication on social media platforms is not only culturally and politically relevant, it is also increasingly widespread across societies. Users not only communicate via social media platforms, but also search specifically for information, disseminate it or post information themselves. However, fake news, hate speech and even radicalizing elements are part of this modern form of communication: Sometimes with far-reaching effects on individuals and societies. A basic understanding of these mechanisms and communication patterns could help to counteract negative forms of communication, e.g. bullying among children or extreme political points of view. To this end, a method will be presented in order to break down the underlying communication patterns, to trace individual users and to inspect their comments and range on social media platforms; Or to contrast them later on via qualitative research. This approeach can identify particularly active users with an accuracy of 100 percent, if the framing social networks as well as the topics are taken into account. However, methodological as well as counteracting approaches must be even more dynamic and flexible to ensure sensitivity and specifity regarding users who spread hate speech, fake news and radicalizing elements.

[4]

arXiv: Do-It-Yourself-Tutorials

- Aktuelle Entwicklungen der Sentiment Analysis lassen sich zum gegenwärtigen Zeitpunkt am besten auf arXiv nachvollziehen



Analysing Social Media Network Data with R

Semi-Automated Screening of Users, Comments and Communication Patterns

a Working Paper

Communication on social media pl
increasingly widespread across societies
search specifically for information, dis
hate speech and even radicalizing elem
with far-reaching effects on individuals
communication patterns could help to
children or extreme political points of
down the underlying communication pa
range on social media platforms; Or to
can identify particularly active users w
well as the topics are taken into accoun
must be even more dynamic and flexib
hate speech, fake news and radicalizing

Sentiment Analysis with R

Natural Language Processing for Semi-Automated Assessments of Qualitative Data

a Working Paper and Tutorial by Dennis Klinkhammer

Abstract

Sentiment analysis is a sub-discipline in the field of natural language processing and computational linguistics and can be used for automated or semi-automated analyses of text documents. One of the aims of these analyses is to recognize an expressed attitude as positive or negative as it can be contained in comments on social media platforms or political documents and speeches as well as fictional and nonfictional texts. Regarding analyses of comments on social media platforms, this is an extension of the previous tutorial on semi-automated screenings of social media network data. A longitudinal perspective regarding social media comments as well as cross-sectional perspectives regarding fictional and nonfictional texts, e.g. entire books and libraries, can lead to extensive text documents. Their analyses can be simplified and accelerated by using sentiment analysis with acceptable inter-rater reliability. Therefore, this tutorial introduces the basic functions for performing a sentiment analysis with R and explains how text documents can be analysed step by step - regardless of their underlying formatting. All prerequisites and steps are described in detail and associated codes are available on GitHub. A comparison of two political speeches illustrates a possible use case.

[4], [5]

arXiv: Do-It-Yourself-Tutorials

- Aktuelle Entwicklungen der Sentiment Analysis lassen sich zum gegenwärtigen Zeitpunkt am besten auf arXiv nachvollziehen

Analysing Social Media Network Data with R

Semi-Automated Screening of Users, Comments and Communication Patterns

a Working Paper

Communication on social media pl
increasingly widespread across societies
search specifically for information, dis
hate speech and even radicalizing elem
with far-reaching effects on individuals
communication patterns could help to
children or extreme political points of
down the underlying communication pat
range on social media platforms; Or to
can identify particularly active users w
well as the topics are taken into accour
must be even more dynamic and flexib
hate speech, fake news and radicalizing

Sentiment Analysis with R

Natural Language Processing for Semi-Automated Assessment

a Working Paper and Tutorial by Dennis Klink

Abstract

Sentiment analysis is a sub-discipline in the field of natural language pr
linguistics and can be used for automated or semi-automated analyses of
aims of these analyses is to recognize an expressed attitude as positive or ne
in comments on social media platforms or political documents and spee
nonfictional texts. Regarding analyses of comments on social media pla
of the previous tutorial on semi-automated screenings of social media n
perspective regarding social media comments as well as cross-sectional persp
nonfictional texts, e.g. entire books and libraries, can lead to extensive text
be simplified and accelerated by using sentiment analysis with acceptable in
this tutorial introduces the basic functions for performing a sentiment ana
text documents can be analysed step by step - regardless of their underlyin
and steps are described in detail and associated codes are available on
political speeches illustrates a possible use case.

Sentiment Analysis: Automatically Detecting Valence, Emotions, and Other Affectual States from Text

Saif M. Mohammad SAIF.MOHAMMAD@NRC-CNRC.GC.CA
National Research Council Canada
1200 Montreal Rd., Ottawa, ON, Canada

Abstract

Recent advances in machine learning have led to computer systems that are human-like in behaviour. Sentiment analysis, the automatic determination of emotions in text, is allowing us to capitalize on substantial previously unattainable opportunities in commerce, public health, government policy, social sciences, and art. Further, analysis of emotions in text, from news to social media posts, is improving our understanding of not just how people convey emotions through language but also how emotions shape our behaviour. This article presents a sweeping overview of sentiment analysis research that includes: the origins of the field, the rich landscape of tasks, challenges, a survey of the methods and resources used, and applications. We also discuss discuss how, without careful fore-thought, sentiment analysis has the potential for harmful outcomes. We outline the latest lines of research in pursuit of fairness in sentiment analysis.

Keywords: *sentiment analysis, emotions, artificial intelligence, machine learning, natural language processing (NLP), social media, emotion lexicons, fairness in NLP*

[4], [5], [7]

arXiv: Do-It-Yourself-Tutorials

- In diesem Beispiel wird ein Text für die Sentiment Analysis vorbereitet

```
text_df <- tibble(line=1:229, text=imported_text$V1)
head(text_df)
```

```
## # A tibble: 6 x 2
##    line text
##   <int> <chr>
## 1     1 "To Sherlock Holmes she is always the woman. I have seldom heard him"
## 2     2 "     mention her under any other name. In his eyes she eclipses and"
## 3     3 "     predominates the whole of her sex. It was not that he felt any"
## 4     4 "     emotion akin to love for Irene Adler. All emotions, and that one"
## 5     5 "     particularly, were abhorrent to his cold, precise but admirably"
## 6     6 "     balanced mind. He was, I take it, the most perfect reasoning and"
```

[5]

arXiv: Do-It-Yourself-Tutorials

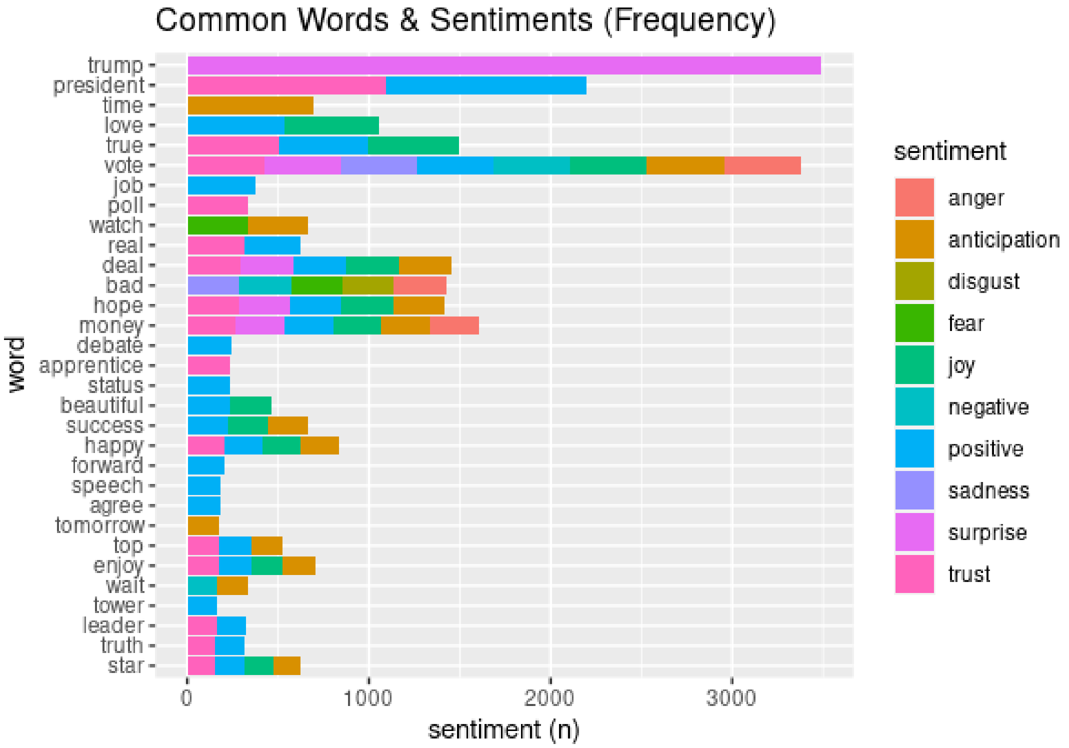
- Darüber hinaus steht eine bedarfsspezifische Ergebnisdarstellung im Vordergrund der Do-It-Yourself-Tutorials

```
nrc_word_counts %>%
  filter(n > 2) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col() +
  coord_flip() +
  labs(y = "sentiment (n)") +
  ggtitle("Common Words & Sentiments (Frequency)")
```

[5]

arXiv: Do-It-Yourself-Tutorials

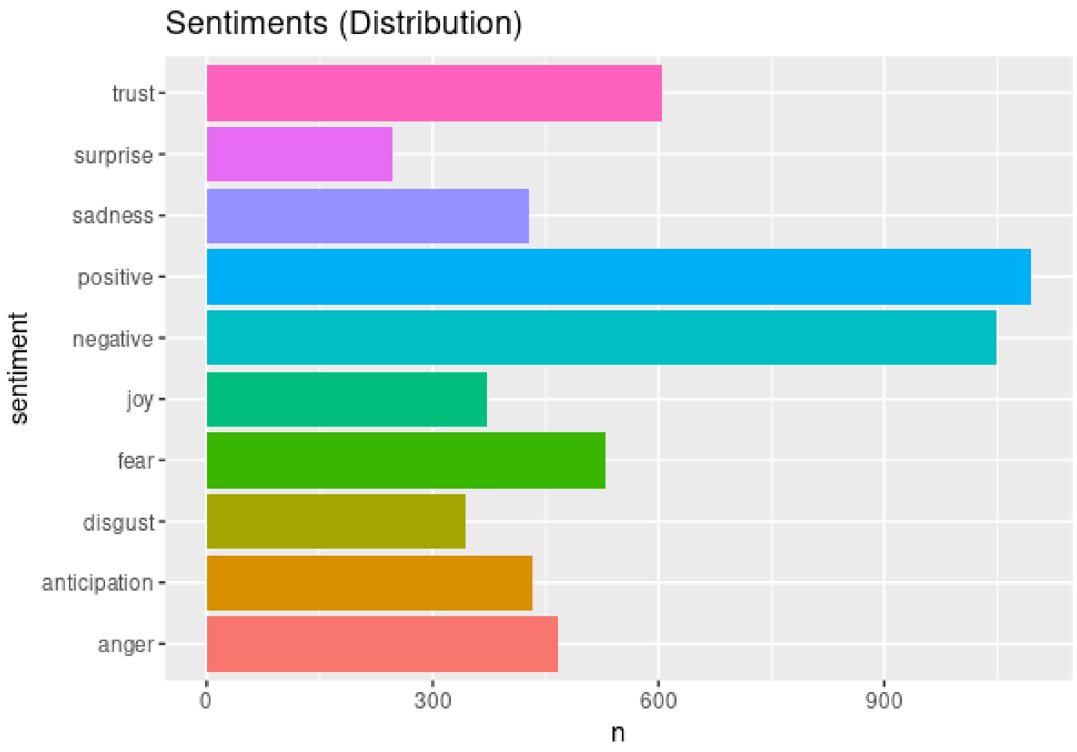
- Beispiel: Sentiment Analysis der **Tweets** von Donald Trump - I



[5]

arXiv: Do-It-Yourself-Tutorials

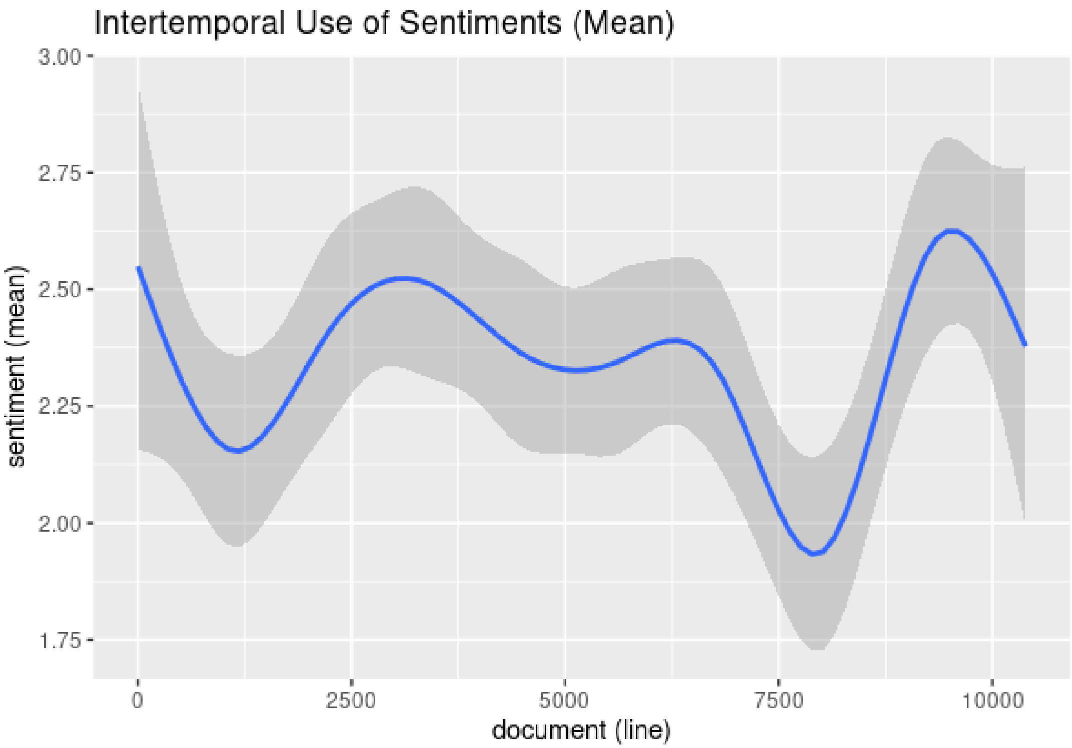
- Beispiel: Sentiment Analysis der **Tweets** von Donald Trump - II



[5]

arXiv: Do-It-Yourself-Tutorials

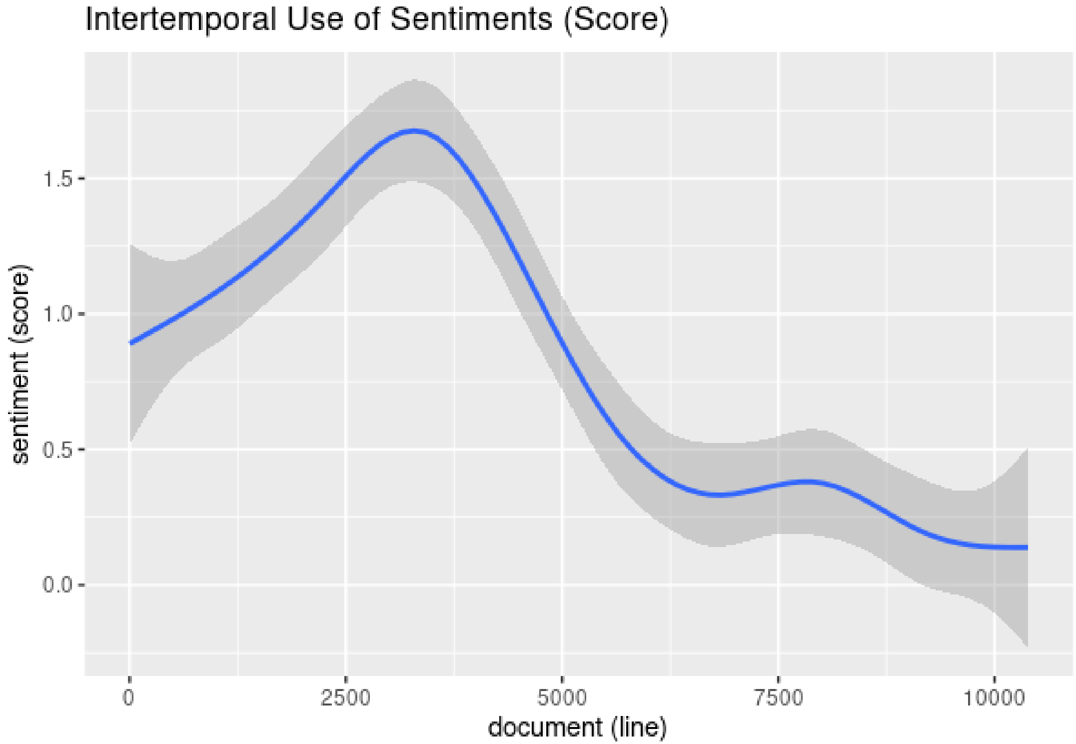
- Beispiel: Sentiment Analysis der **Tweets** von Donald Trump - III



[5]

arXiv: Do-It-Yourself-Tutorials

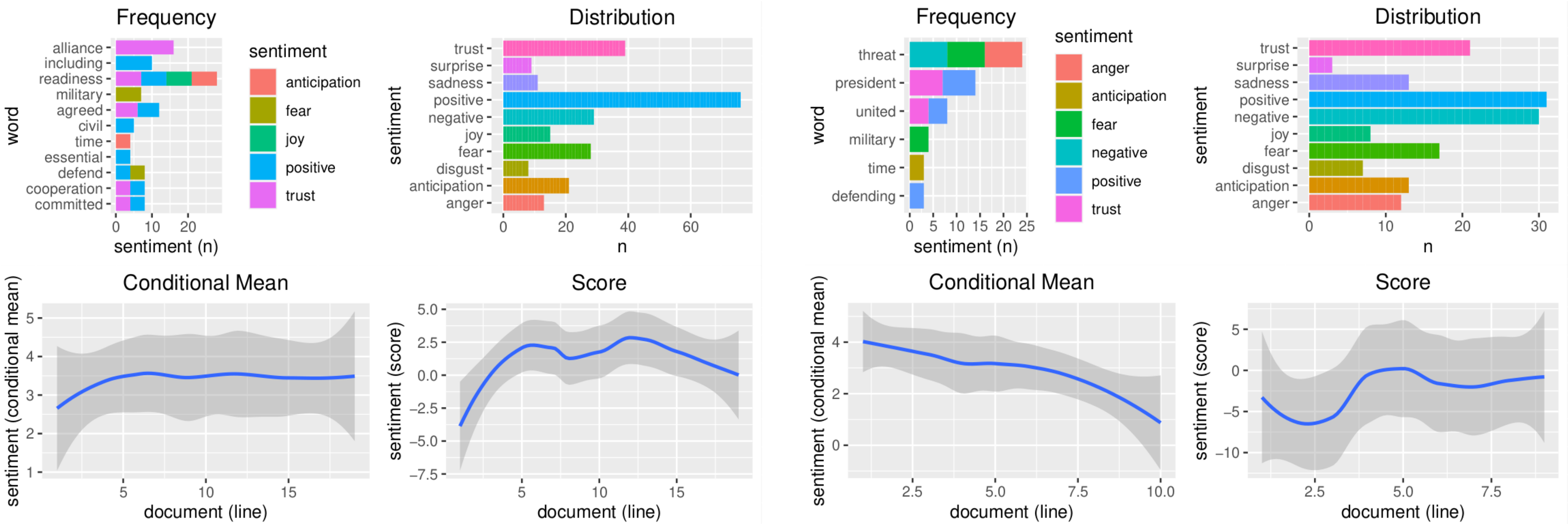
- Beispiel: Sentiment Analysis der **Tweets** von Donald Trump - IV



[5]

arXiv: Do-It-Yourself-Tutorials

- Ein sprachlicher Vergleich zwischen **Stoltenberg** (l.) und **Lawrow** (r.)



[5]

Fallstricke der Sentiment Analysis

Fallstricke der Sentiment Analysis

- Übernahme von bestehenden Tools, ohne Verständnis der zugrundeliegenden **Programmiersprache**

[1]

Fallstricke der Sentiment Analysis

- Übernahme von bestehenden Tools, ohne Verständnis der zugrundeliegenden **Programmiersprache**
- Keine bzw. unzureichende Kenntnis der Funktionsweise von Natural Language Processing (**Deep Learning**, insb. Aktivierungsfunktionen)

[1], [6]

Fallstricke der Sentiment Analysis

- Übernahme von bestehenden Tools, ohne Verständnis der zugrundeliegenden **Programmiersprache**
- Keine bzw. unzureichende Kenntnis der Funktionsweise von Natural Language Processing (**Deep Learning**, insb. Aktivierungsfunktionen)
- Keine bzw. unzureichende Berücksichtigung des geographischen und (!) zeitlichen **Kontextes**

[1], [6], [7]

Fallstricke der Sentiment Analysis

- Übernahme von bestehenden Tools, ohne Verständnis der zugrundeliegenden **Programmiersprache**
- Keine bzw. unzureichende Kenntnis der Funktionsweise von Natural Language Processing (**Deep Learning**, insb. Aktivierungsfunktionen)
- Keine bzw. unzureichende Berücksichtigung des geographischen und (!) zeitlichen **Kontextes**
- Rückgriff auf nicht aktualisierte bzw. unspezifische **Lexika**

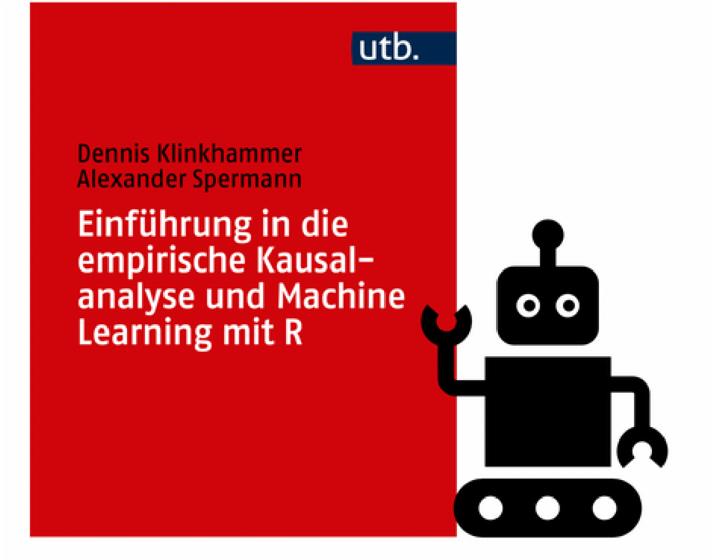
[1], [6], [7], [2]

Fallstricke der Sentiment Analysis

- Übernahme von bestehenden Tools, ohne Verständnis der zugrundeliegenden **Programmiersprache**
- Keine bzw. unzureichende Kenntnis der Funktionsweise von Natural Language Processing (**Deep Learning**, insb. Aktivierungsfunktionen)
- Keine bzw. unzureichende Berücksichtigung des geographischen und (!) zeitlichen **Kontextes**
- Rückgriff auf nicht aktualisierte bzw. unspezifische **Lexika**
- **Transfer Learning** nur eingeschränkt möglich (!)

[1], [6], [7], [2]

Vielen Dank für die Aufmerksamkeit



Statistical Thinking

Mehr Beispiele zur Statistik mit Python und R

Referenzen

(1) Grogan, M. (2020): *NLP from a time series perspective. How time series analysis can complement NLP*. Towards Data Science.

(2) Hamachers, A., Weber, K., Widmann, J. & S. Jarolimek (2020): *Extremistische Dynamiken im Social Web*. Frankfurt am Main: Verlag für Polizeiwissenschaft.

(3) Iacus, S. M. & G. Porro (2022): *Using social networks to measure subjective well-being*. In: Significance. Volume 19. Issue 3.

(4) Klinkhammer, D. (2020): *Analysing Social Media Network Data with R: Semi-Automated Screening of Users, Comments and Communication Patterns*. Cornell University (arXiv).

(5) Klinkhammer, D. (2022): *Sentiment Analysis with R: Natural Language Processing for Semi Automated Assessments of Qualitative Data*. Cornell University (arXiv).

(6) Ng, A. (2022): Deep Learning Specialization. DeepLearning.AI, Palo Alto, CA (USA).

(7) Mohammad, S. M. (2021): *Sentiment Analysis: Automatically Detecting Valence, Emotions and Other Affectual States from Text*. Cornell University (arXiv).