



## Session 3: **Diktionäre**



# „Magie“ verstehen: Klassische Schritte

1. Preprocessing

2. Analyse

3. Test auf  
Qualitätskriterien



**Session 1**  
(Einführung &  
Preprocessing)



**Session 2-4**  
(Co-Occurrence,  
Diktionäre,  
Topic Modeling)



**Session 5**  
(Qualitätskriterien)



# Agenda

---

1. Einführung
2. Off-the-Shelf Diktionäre
3. Organische Diktionäre
4. Anwendungsbeispiel in der Kowi
5. Outro

# 1. Einführung

# Diktionäre

---

- Diktionäre sind Listen von Features, die ein bestimmtes Konstrukt (z.B. Emotionalisierung) beschreiben.
- In Diktionär-Analysen zählen wir, wie häufig *manifeste Features* vorkommen, darauf zu schliessen, inwiefern ein *latentes Konstrukt* vorkommt.



Mit welchen Features würdet  
ihr das latente Konstrukt  
«emotionale Sprache» messen?



# Diktionäre

---

- Diktionäre sind Listen von Features, die ein bestimmtes Konstrukt (z.B. Emotionalisierung) beschreiben.
- In Diktionär-Analysen zählen wir, wie häufig *manifeste Features* vorkommen, darauf zu schliessen, inwiefern ein *latentes Konstrukt* vorkommt.
- Deduktives Verfahren: Texte in vorgegebene Kategorien klassifizieren



# Klassisches Beispiel: Sentiment-Analyse

---

*“the idea that the affective content of text [...] reveals information about the underlying opinions, stances, and attitudes”* (Rauh, [2018](#), S. 320)

- Erfassung von Sentiment (z. B. Vorkommen von Features, die mit negativem bzw. Positivem Sentiment assoziiert werden) (Hase, [2021](#); Stine, [2019](#); Taboada, [2016](#))
- Ziel ist es, Meinungen, Evaluationen, Einstellung zu identifizieren (????)



# Diktionäre

---

1. Preprocessing
2. Diktionär wählen
3. Analyse
4. Validierung



# Diktionäre

---

1. **Preprocessing**
2. Diktionär wählen/erstellen
3. Analyse
4. Validierung



## Wichtige Fragen u. a.:

1. Soll ich Stoppwörter entfernen (oder nicht)?
2. Soll ich Stemming etc. nutzen (oder nicht)?

# Diktionäre

1. Preprocessing
2. **Diktionär wählen/erstellen**
3. Analyse
4. Validierung



## Wichtige Fragen u. a.:

1. Benutze ich ein Off-the-Shelf Diktionär oder ein organisches Diktionär? (*Art Diktionär*)
2. Wie breit bzw. spezifisch soll mein Diktionär sein? (*Anzahl Features Diktionär*)



# „Off-the-shelf“ vs. organische Diktionäre

---

- **Off-the-shelf:** Rückgriff auf bestehende Wortlisten (etwa für Emotionen, Sentiment oder Themen), oft entwickelt in anderen Kontexten
- **Organisch:** Eigene, domänspezifische Wortlisten

# Breite vs. spezifische Diktionäre

---

- **Breite Diktionäre:** möglichst umfassende Wortlisten (d.h. viele Features; ggf. auch solche, die latentes Konstrukt nicht besonders präzise fassen)
- **Spezifische Diktionäre:** möglichst spezifische Wortlisten (d.h. wenige Features; nur solche, die latentes Konstrukt besonders präzise fassen)



Welche Konsequenzen kann die Wahl eines breiten  
vs. spezifischen Diktionärs haben?



# Breite vs. spezifische Diktionäre

---

- **Breite Diktionäre:** möglichst umfassende Wortlisten (d.h. viele Features; ggf. auch solche, die latentes Konstrukt nicht besonders präzise fassen:  
**hoher Recall, niedrige Präzision; weiter Sitzung 5)**
- **Spezifische Diktionäre:** möglichst spezifische Wortlisten (d.h. wenige Features; nur solche, die latentes Konstrukt besonders präzise fassen:  
**niedriger Recall, hohe Präzision; weiter Sitzung 5)**

# Diktionäre

1. Preprocessing
2. Diktionär wählen/erstellen
3. **Analyse**
4. Validierung



## Wichtige Fragen u. a.:

1. Wie garantiere ich, dass sich z.B. Sentiment auf ein spezifisches Objekt (z.B. ein Thema) bezieht?

→ Suche nach Features im gesamten Text vs. nur Features bezogen auf ein Thema/Person/Ort, etc.

# Diktionäre

---

1. Preprocessing
2. Diktionär wählen/erstellen
3. **Analyse**
4. Validierung



## Wichtige Fragen u. a.:

1. Wie garantiere ich, dass sich z.B. Sentiment auf ein spezifisches Objekt (z.B. ein Thema) bezieht?
  2. Wie gehe ich z. B. mit Negierung um?
- Was ist mit Ausdrücken wie not bad?

# Diktionäre

1. Preprocessing
2. Diktionär wählen/erstellen
3. **Analyse**
4. Validierung



## Wichtige Fragen u. a.:

1. Wie garantiere ich, dass sich z.B. Sentiment auf ein spezifisches Objekt (z.B. ein Thema) bezieht?
2. Wie gehe ich z. B. mit Negierung um?
3. Wie gehe ich z. B. mit unterschiedlich langen Texten um?

→ Längere Texte haben eine höhere Chance “zufällig” z. B. negative Features zu enthalten.



# Diktionäre

1. Preprocessing
2. Diktionär wählen/erstellen
3. **Analyse**
4. Validierung



## Wichtige Fragen u. a.:

1. Wie garantiere ich, dass sich z.B. Sentiment auf ein spezifisches Objekt (z.B. ein Thema) bezieht?
2. Wie gehe ich z. B. mit Negierung um?
3. Wie gehe ich z. B. mit unterschiedlich langen Texten um?
4. Ab wann gelten latente Konstrukte als identifiziert?

→ Einmaliges Feature-Vorkommen? Mehr?

# Diktionäre

---

1. Preprocessing
2. Diktionär wählen/erstellen
3. Analyse
4. **Validierung**



**Sitzung 5!**



# Validität bei Diktionären: ein Beispiel

---

**Pleased** to report this machine made the **most delicious** coffee I have had from a home brew machine. **Very best** from the first time.

vs.

Have to push its **fancy** button over and over to get a pot of coffee. **Lots of fun** first thing in the morning.  
**Not a great** way to start the day.

## 2. Off-the-Shelf Diktionäre



# „Off-the-shelf“-Diktionäre

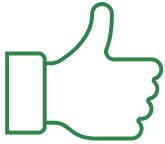
---

Rückgriff auf bestehende Wortlisten (etwa für Emotionen, Sentiment oder Themen), u.a.:

- [General Inquirer](#) (GI)
- [Linguistic Inquiry and Word Count](#) (LWIC)
- [Lexicoder Sentiment Dictionary](#) (LSD)
- [WordNet-Affect](#) (WNA)
- [Bing](#)
- [AFINN](#)
- [NRC Word-Emotion Association Lexicon](#) (EmoLex)
- [SentimentWortschatz](#) (SentiWS)

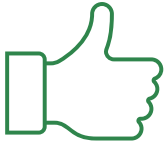
# Vor- & Nachteile von Off-the-Shelf Diktionären

---

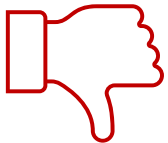


- Große Auswahl aus bestehenden Diktionären: kostengünstig
- Oft bereits in Paketen wie `quanteda` implementiert
- Transparent, d.h. kein „black-box“-Verfahren

# Vor- & Nachteile von Off-the-Shelf Diktionären



- Große Auswahl aus bestehenden Diktionären: kostengünstig
- Oft bereits in Paketen wie `quanteda` implementiert
- Transparent, d.h. kein „black-box“-Verfahren



- Oft *a-theoretisch und wenig valide*, da nicht von Kommunikationswissenschaftler:innen für unsere Theorien/Daten entwickelt

# Für welche Fragestellungen eignen sich solche Verfahren?

- Textanalyse, u. a.: Analyse von Nachrichtenfaktoren wie „Negativität“ (Burggraaff & Trilling, 2020)

Journalism  
Volume 21, Issue 1, January 2020, Pages 112-129  
© The Author(s) 2017, Article Reuse Guidelines  
<https://doi.org/10.1177/1464884917716699>



Article



## Through a different gate: An automated content analysis of how online news and print news differ

Christiaan Burggraaff<sup>1</sup> and Damian Trilling<sup>2</sup>

### Abstract

We investigate how news values differ between online and print news articles. We hypothesize that print and online articles differ in terms of news values because of differences in the routines used to produce them. Based on a quantitative automated content analysis of  $N = 762,095$  Dutch news items, we show that online news items are more likely to be follow-up items than print items, and that there are further differences regarding news values like references to persons, the power elite, negativity, and positivity. In order to conduct this large-scale analysis, we developed innovative methods to automatically code a wide range of news values. In particular, this article demonstrates how techniques such as sentiment analysis, named entity recognition, supervised machine learning, and automated queries of external databases can be combined and used to study journalistic content. Possible explanations for the difference found between online and offline news are discussed.





# Wie kann ich diese Analysen in R anwenden?

---

- Kleine Auswahl möglicher R-Pakete
  - „base R“ (z.B. für Identifikation von Features via `grepl()` etc,)
  - „quanteda“ bzw. „quanteda.sentiment“ (z.B. für Off-the-Shelf-Diktionäre)
  - „tidytext“ (z.B. für Off-the-Shelf-Diktionäre)

```
function(scope, element, attr, ngSwitchController) {  
  var watchExpr = attr.ngSwitch || attr.on,  
      selectedTranscludes = [],  
      selectedElements = [],  
      previousElements = [],  
      selectedScopes = [];  
  
  scope.$watch(watchExpr, function ngSwitchWatchAction(value) {  
    var i, ii;  
    for (i = 0, ii = previousElements.length; i < ii; ++i) {  
      previousElements[i].remove();  
    }  
    previousElements.length = 0;  
  
    for (i = 0, ii = selectedScopes.length; i < ii; ++i) {  
      var selected = selectedElements[i];  
      selectedScope[i].destroy();  
    }  
  });  
  
  selectedElements.length = 0;  
  selectedScopes.length = 0;  
  
  if ((selectedTransclude = ...))
```

Zeit für R!

```
selectedElements.length = 0;  
selectedScopes.length = 0;  
  
if ((selectedTransclude = ...))
```

# Pakete installieren & aktivieren

---

```
#install.packages("tidyverse")  
#install.packages("RCurl")  
#install.packages("quanteda")  
  
library("tidyverse")  
library("RCurl")  
library("quanteda")
```

# Daten einlesen & Preprocessing

```
# Daten laden
url <- getURL("https://raw.githubusercontent.com/valeriehase/textasdata-ms/main/data/
data <- read.csv2(text = url)

# Preprocessing
tokens <- tokens(data$Description,
                  what = "word", #Tokenisierung, hier zu Wörtern als Analyseeinheit
                  remove_punct = TRUE, #Entfernung von Satzzeichen
                  remove_numbers = TRUE) %>% #Entfernung von Zahlen

# Kleinschreibung
tokens_tolower()

# Text-as-Data Repräsentation als Document-Feature-Matrix
dfm <- tokens %>%
  dfm()
```

# Diktionär wählen

```
# Wir schauen uns das Diktionär an
data_dictionary_LSD2015 %>%
  head()
```

Dictionary object with 4 key entries.

- [negative]:
  - a lie, abandon\*, abas\*, abattoir\*, abdicat\*, aberras\*, abhor\*, abject\*, abnormal\*, a
- [positive]:
  - ability\*, abound\*, absolv\*, absorbent\*, absorption\*, abundanc\*, abundant\*, acced\*,
- [neg\_positive]:
  - best not, better not, no damag\*, no no, not ability\*, not able, not abound\*, not ab
- [neg\_negative]:
  - not a lie, not abandon\*, not abas\*, not abattoir\*, not abdicat\*, not aberras\*, not a

# Features aus Diktionär identifizieren

```
sentiment_tvshows <- dfm %>%  
  
  # Suche nach Features aus Diktionär  
  # Gewichtung relativ zur Anzahl aller Wörter  
  dfm_weight(scheme = "prop") %>%  
  dfm_lookup(dictionary = data_dictionary_LSD2015[1:2])  
  
# Ausgabe der Ergebnisse  
sentiment_tvshows %>%  
  head()
```

# Features aus Diktionär identifizieren

Document-feature matrix of: 6 documents, 2 features (8.33% sparse) and 0 docvars.

	features	
docs	negative	positive
text1	0.09523810	0.04761905
text2	0.04000000	0.04000000
text3	0.13636364	0.04545455
text4	0	0.04545455
text5	0.11538462	0.11538462
text6	0.02857143	0.05714286

Wir sehen z. B. für die allerste Beobachtung, die Beschreibung von *Game of Thrones*:

```
data$Description[1]
```

```
[1] "Nine noble families fight for control over the lands of Westeros, while an ancient
```

# Serien als „negativ“, „neutral“, oder „positiv“ klassifizieren

```
# Ergebnis für die weitere Analyse in einen Data Frame umwandeln
sentiment_tvshows <- convert(sentiment_tvshows,
                             to = "data.frame") %>%

# Umwandlung in tibble-Format
as_tibble %>%

# Wir ergänzen zunächst wieder die Serientitel & das "Parental-Rating"
mutate(Title = data$Title,
       Parental.Rating = data$Parental.Rating) %>%

# Wir erstellen eine Variable, die Texte als
# "neutral", "positiv" oder "negativ" identifiziert

# Zunächst gelten alle Texte als "neutral"
mutate(sentiment = "neutral",

       # Falls mehr pos. als neg: "positiv"
       sentiment = replace(sentiment,
                           positive > negative,
                           "positiv"),

       # Falls mehr neg. als pos.: "negativ"
       sentiment = replace(sentiment,
                           positive < negative,
                           "negativ")) %>%
```

```
# Sortierung der Variablen
select(Title, Parental.Rating, positive, negative, sentiment)

# Ausgabe des Ergebnis
sentiment_tvshows %>%
  head()
```



# Serien als „negativ“, „neutral“, oder „positiv“ klassifizieren

```
# A tibble: 6 × 5
  Title                Parental.Rating positive negative sentiment
  <chr>                <chr>          <dbl>    <dbl> <chr>
1 1. Game of Thrones TV-MA            0.0476  0.0952 negativ
2 2. Breaking Bad    TV-MA            0.04     0.04 neutral
3 3. Stranger Things TV-14            0.0455  0.136 negativ
4 4. Friends         TV-14            0.0455  0      positiv
5 5. The Walking Dead TV-MA            0.115   0.115 neutral
6 6. Sherlock        TV-14            0.0571  0.0286 positiv
```

# Serien als „negativ“, „neutral“, oder „positiv“ klassifizieren

```
# Anzahl neutral, negativer und positiver Texte?  
sentiment_tvshows %>%
```

```
# absolute Anzahl jeder Sentiment-Art (n)  
count(sentiment) %>%
```

```
# Ausgabe in Prozent (perc)  
mutate(perc = prop.table(n)*100,  
       perc = round(perc, 2))
```

```
# A tibble: 3 x 3  
  sentiment      n  perc  
  <chr>      <int> <dbl>  
1 negativ     406  45.1  
2 neutral     234   26  
3 positiv     260  28.9
```

# Was sind die negativsten/positivsten Serien?

```
# Negativste Serien
sentiment_tvshows %>%
  arrange(desc(negative)) %>%
  slice(1:5)
```

```
# A tibble: 5 × 5
```

	Title	Parental.Ra
	<chr>	<chr>
1	480. Leverage	TV-PG
2	824. The Glory	TV-MA
3	115. 24	TV-14
4	205. Revenge	TV-14
5	284. Falling Skies	TV-14

```
# Positivste Serien
```

```
sentiment_tvshows %>%
  arrange(desc(positive)) %>%
  slice(1:5)
```

```
# A tibble: 5 × 5
```

	Title	Parental.Rating	positive	negative	sentiment
	<chr>	<chr>	<dbl>	<dbl>	<chr>
1	262. Dead to Me	TV-MA	0.238	0.0952	positiv
2	510. Ghost Whisperer	TV-PG	0.226	0.0323	positiv
3	704. The Bugs Bunny Show	TV-G	0.222	0	positiv
4	531. Ugly Betty	TV-PG	0.211	0.0526	positiv
5	532. Coupling	TV-14	0.211	0	positiv

# Sind Serien mit "höherem" Parental-Rating negativer?

```
sentiment_tvshows <- sentiment_tvshows %>%  
  
#Erstellen eines "Rating.Adults"-Klassifizierungs-Variable  
mutate(Rating.Adults = 0,  
       Rating.Adults = replace(Rating.Adults,  
                               Parental.Rating == "TV-MA",  
                               1))
```

```
#Wir schauen uns  
head(sentiment_tvshows)
```

```
# A tibble: 6 x 6
```

	Title	Parental.Rating	positive	negative	sentiment	Rating.Adults
	<chr>	<chr>	<dbl>	<dbl>	<chr>	<dbl>
1	1. Game of Thrones	TV-MA	0.0476	0.0952	negativ	1
2	2. Breaking Bad	TV-MA	0.04	0.04	neutral	1
3	3. Stranger Things	TV-14	0.0455	0.136	negativ	0
4	4. Friends	TV-14	0.0455	0	positiv	0
5	5. The Walking Dead	TV-MA	0.115	0.115	neutral	1
6	6. Sherlock	TV-14	0.0571	0.0286	positiv	0

# Visualisierung

```
plot <- sentiment_tvshows %>%
```

```
# Wir berechnen die gruppierten Häufigkeiten
group_by(Rating.Adults) %>%
```

```
# absolute Anzahl jeder Sentiment-Art
count(sentiment) %>%
```

```
# Ausgabe in Prozent (perc)
mutate(perc = prop.table(n)*100,
       perc = round(perc, 2)) %>%
```

```
# Wir heben die Gruppierung auf
ungroup()
```

```
# Visualisierung
```

```
ggplot(plot, aes(fill = Rating.Adults, y = perc, x = sentiment)) +
```

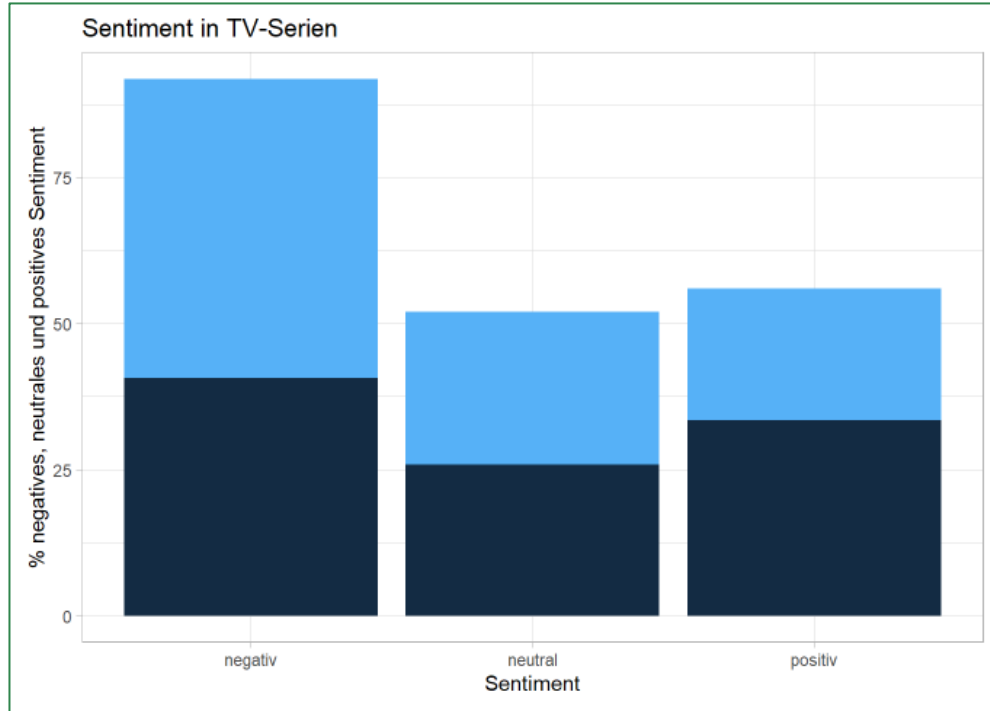
```
# Wir kreieren den entsprechenden Graphen
geom_bar(stat = "identity") +
```

```
# Wir fügen Achsenbeschriftungen hinzu
labs(y = "% negatives, neutrales und positives Sentiment",
     x = "Sentiment",
     title = "Sentiment in TV-Serien",
     colour = "Sentiment") +
```

```
# Wir ändern das Background-Design
theme_light() +
```

```
# Wir entfernen die Legende, die wenig hilfreich ist
theme(legend.position = "none")
```

# Visualisierung





Pause

### 3. Organische Diktionäre





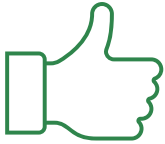
# Organische Diktionäre

---

- **Organische Diktionäre:** Eigene, domänspezifische Wortlisten
- **Schritte**, um organische Diktionäre zu erstellen (Muddiman et al., [2019](#); Stoll et al., [2023](#)):
  - Feature-Identifikation durch Rückgriff auf existierende Studien
  - Erweiterung auf Basis des Korpus
    - z. B. Inspektion von Top Features
    - z.B. automatisierte Identifikation weiterer Features durch Word Embeddings
  - Validierung jedes einzelnen Features
  - Validierung des gesamten Diktionärs

# Vor- & Nachteile von organischen Diktionären

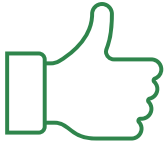
---



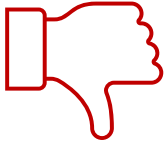
- Oftmals enger an theoretische Konzepte und damit valider gebunden
- Transparent, d.h. kein „black-box“-Verfahren

# Vor- & Nachteile von organischen Diktionären

---



- Oftmals enger an theoretische Konzepte und damit valider gebunden
- Transparent, d.h. kein „black-box“-Verfahren



- Erstellung sehr aufwendig (Zeit, Codierer:innen)
- Wenig bis gar nicht generalisierbar, da eng an Korpus gebunden

# Für welche Fragestellungen eignen sich solche Verfahren?

- Textanalyse, u. a.: Inzivilität auf digitalen Plattformen (Stoll et al., [2023](#))





# Wie kann ich diese Analysen in R anwenden?

---

- Kleine Auswahl möglicher R-Pakete
  - „base R“ (z.B. für Identifikation von Features via `grepl()` etc,)
  - „quanteda“ (z.B. für Erstellung eigener Diktionäre)

```
function(scope, element, attr, ngSwitchController) {  
  var watchExpr = attr.ngSwitch || attr.on,  
      selectedTranscludes = [],  
      selectedElements = [],  
      previousElements = [],  
      selectedScopes = [];  
  
  scope.$watch(watchExpr, function ngSwitchWatchAction(value) {  
    var i, ii;  
    for (i = 0, ii = previousElements.length; i < ii; ++i) {  
      previousElements[i].remove();  
    }  
    previousElements.length = 0;  
  
    for (i = 0, ii = selectedScopes.length; i < ii; ++i) {  
      var selected = selectedElements[i];  
      selectedScope[i].destroy();  
    }  
  });  
  
  selectedElements.length = 0;  
  selectedScopes.length = 0;  
  
  if ((selectedTransclude = ...))
```

Zeit für R!

# Diktionär erstellen

---

```
diktionär_crime <- dictionary(list(crime = c("crim*", "police*", "gun*",  
                                             "shot*", "dead*", "murder*",  
                                             "kill*", "court*", "suspect*",  
                                             "witness*", "arrest*", "officer*",  
                                             "verdict*"))))
```

# Analyse

```
#Diktionär anwenden
crime_tvshows <- dfm %>%
  dfm_weight(scheme = "prop") %>%
  dfm_lookup(dictionary = dktionär_crime)

# Ergebnis für die weitere Analyse in einen Data Frame
crime_tvshows <- convert(crime_tvshows,
  to = "data.frame") %>%

# Umwandlung in tibble-Format
as_tibble %>%

# Wir ergänzen zunächst wieder die Serientitel
mutate(Title = data$Title) %>%
```

```
# Wir erstellen eine Variable, die Texte als
# "crime" oder "non-Crime" identifiziert
mutate(crime_binary = "crime",
  crime_binary = replace(crime_binary,
    crime == 0,
    "non-crime")) %>%

# Sortierung der Variablen
select(Title, crime, crime_binary)

#Ausgabe der Ergebnisse
head(crime_tvshows)
```



# Analyse

```
# A tibble: 6 × 3
  Title                crime crime_binary
  <chr>                <dbl> <chr>
1 1. Game of Thrones      0 non-crime
2 2. Breaking Bad         0 non-crime
3 3. Stranger Things      0 non-crime
4 4. Friends              0 non-crime
5 5. The Walking Dead     0 non-crime
6 6. Sherlock             0 non-crime
```

# Wie viele Serien sind Krimis?

```
#Ausgabe der Crime vs. Non-Crime Serien
crime_tvshows %>%

# absolute Anzahl jeder Sentiment-Art (n)
count(crime_binary) %>%

# Ausgabe in Prozent (perc)
mutate(perc = prop.table(n)*100,
       perc = round(perc, 2))

# A tibble: 2 x 3
  crime_binary      n  perc
  <chr>         <int> <dbl>
1 crime          170  18.9
2 non-crime       730  81.1
```

# Welche Serien sind am „klarsten“ Krimis?

```
crime_tvshows %>%  
  arrange(desc(crime)) %>%  
  slice(1:5)
```

```
# A tibble: 5 × 3
```

	Title <chr>	crime <dbl>	crime_binary <chr>
1	742. Behzat Ç: An Ankara Detective Story	0.267	crime
2	815. Southland	0.25	crime
3	264. American Crime Story	0.182	crime
4	162. Bodyguard	0.176	crime
5	314. Bosch	0.154	crime

## 4. Anwendungsaufgabe



# Jetzt seid ihr dran!



Könnt ihr...



Basis: Analysieren, wie viel Prozent der Serien Science-Fiction Serien sind?



Fortgeschritten: Analysieren, welche fünf Science-Fiction Serien die (nach Publikums-Votum laut „`Number.of.Votes`“) beliebtesten Serien sind?

# Jetzt seid ihr dran!



```
#Diktionär erstellen
diktionär_scifi <- dictionary(list(scifi = c("fantas*", "science fict*", "sci-fi*",
      "sci fi*", "galax*", "space*",
      "magic*", "wizzard*", "witch*",
      "monster")))
```

# Jetzt seid ihr dran!



```
#Diktionär anwenden
scifi_tvshows <- dfm %>%
  dfm_weight(scheme = "prop") %>%
  dfm_lookup(dictionary = dktionär_scifi)
```

```
# Ergebnis für die weitere Analyse in
scifi_tvshows <- convert(scifi_tvshow
                        to = "data.f
```

```
# Umwandlung in tibble-Format
as_tibble %>%
```

```
# Wir ergänzen zunächst wieder die
mutate(Title = data$Title,
       Number.of.Votes = data$Numbe
```

```
# Wir erstellen eine Variable, die Texte als
# "scifi" oder "non-scifi" identifiziert
```

```
mutate(scifi_binary = "scifi",
       scifi_binary = replace(scifi_binary,
                             scifi == 0,
                             "non-scifi")) %>%
```

```
# Sortierung der Variablen
select(Title, Number.of.Votes, scifi, scifi_binary)
```

```
#Ausgabe der Ergebnisse
head(scifi_tvshows)
```

# Jetzt seid ihr dran!



```
# Wie viel Prozent sind Sci-Fi Serien?  
scifi_tvshows %>%  
  
# absolute Anzahl jeder Sentiment-Art (n)  
count(scifi_binary) %>%  
  
# Ausgabe in Prozent (perc)  
mutate(perc = prop.table(n)*100,  
       perc = round(perc, 2))  
  
# A tibble: 2 × 3  
  scifi_binary      n perc  
  <chr>         <int> <dbl>  
1 non-scifi      855    95  
2 scifi          45     5
```



# Jetzt seid ihr dran!



```
#Diktionär erstellen
scifi_tvshows %>%

#Auswahl der Sci-Fi-Serien
filter(scifi_binary == "scifi") %>%

#Umwandlung der Number.of.Votes Variable
mutate(Number.of.Votes = gsub("K", "000000", Number.of.Votes),
       Number.of.Votes = gsub("M", "0000000000", Number.of.Votes),
       Number.of.Votes = as.numeric(Number.of.Votes)) %>%

#Sortierung nach Rating
arrange(desc(Number.of.Votes)) %>%

#Ausgabe der ersten fünf Serien
head(5)
```

# Jetzt seid ihr dran!



```
# A tibble: 5 × 4
  Title                Number.of.Votes  scifi scifi_binary
<chr>                <dbl>    <dbl> <chr>
1 18. The Mandalorian  590000000 0.0455 scifi
2 21. The Witcher      572000000 0.0435 scifi
3 70. Firefly          282000000 0.0526 scifi
4 74. The 100          277000000 0.0385 scifi
5 90. Doctor Who       248000000 0.0476 scifi
```



Jetzt seid ihr dran!

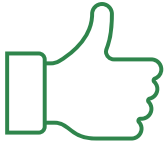


Wie würdet ihr herausfinden, welche Serien falsch bzw. richtig klassifiziert wurden, um eure Analysen zu überprüfen bzw. verbessern?

## 5. Outro

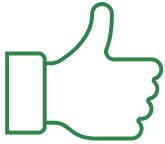
# Vor- und Nachteile von Diktionären

---

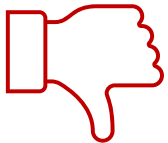


- Relativ einfach anzuwenden, d.h. keine fortgeschrittenen R- bzw. Statistik-Kenntnisse notwendig
- Große Auswahl aus bestehenden Diktionären
- reliabel & reproduzierbar: wiederholte Messungen ergeben gleiche Ergebnisse

# Vor- und Nachteile von Diktionären



- Relativ einfach anzuwenden, d.h. keine fortgeschrittenen R- bzw. Statistik-Kenntnisse notwendig
- Große Auswahl aus bestehenden Diktionären
- reliabel & reproduzierbar: wiederholte Messungen ergeben gleiche Ergebnisse



- Welche theoretischen Konstrukte können wir damit messen? („Sentiment“ ist kein theoretisches Konstrukt!)
- Große Zweifel an Validität, z.B. im Vergleich zu maschinellem Lernen

# Kritische Einschätzung: We have moved on!

COMMUNICATION METHODS AND MEASURES  
2020, VOL. 14, NO. 2, 83–104  
<https://doi.org/10.1080/19312458.2019.1671966>

 Routledge  
Taylor & Francis Group

 OPEN ACCESS  Check for updates

## What's the Tone? Easy Doesn't Do It: Analyzing Performance and Agreement Between Off-the-Shelf Sentiment Analysis Tools

Mark Boukes , Bob van de Velde, Theo Araujo , and Rens Vliegenthart

Amsterdam School of Communication Research (ASCoR), University of Amsterdam, the Netherlands


**ABSTRACT**  
This article scrutinizes the method of automated content analysis to measure the tone of news coverage. We compare a range of off-the-shelf sentiment analysis tools to manually coded economic news as well as examine the agreement between these dictionary approaches themselves. We assess the performance of five off-the-shelf sentiment analysis tools and two tailor-made dictionary-based approaches. The analyses result in five conclusions. First, there is little overlap between the off-the-shelf tools; causing wide divergence in terms of tone measurement. Second, there is no stronger overlap with manual coding for short texts (i.e., headlines) than for long texts (i.e., full articles). Third, an approach that combines individual dictionaries achieves a comparably good performance. Fourth, precision may increase to acceptable levels at higher levels of granularity. Fifth, performance of dictionary approaches depends more on the number of relevant keywords in the dictionary than on the number of valenced words as such; a small tailor-made lexicon was not inferior to large established dictionaries. Altogether, we conclude that off-the-shelf sentiment analysis tools are mostly unreliable and unsuitable for research purposes – at least in the context of Dutch economic news – and manual validation for the specific language, domain, and genre of the research project at hand is always warranted.



## Klassische Probleme (Boukes et al., [2020](#)):

- Schwache/keine Korrelation von Diktionären mit manueller Codierung
- Schwache/keine Korrelation von Diktionären untereinander




# Kritische Einschätzung: We have moved on!

COMMUNICATION METHODS AND MEASURES  
2021, VOL. 15, NO. 2, 121–140  
<https://doi.org/10.1080/19312458.2020.1869198>

 **Routledge**  
Taylor & Francis Group

 OPEN ACCESS  Check for updates

## The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms

Wouter van Atteveldt , Mariken A. C. G. van der Velden , and Mark Boukes 

<sup>a</sup>Department of Communication Science, Vrije Universiteit Amsterdam; <sup>b</sup>Department of Communication Science, Amsterdam School of Communications Research (ASCoR), University of Amsterdam

**ABSTRACT**  
Sentiment is central to many studies of communication science, from negativity and polarization in political communication to analyzing product reviews and social media comments in other sub-fields. This study provides an exhaustive comparison of sentiment analysis methods, using a validation set of Dutch economic headlines to compare the performance of manual annotation, crowd coding, numerous dictionaries and machine learning using both traditional and deep learning algorithms. The three main conclusions of this article are that: (1) The best performance is still attained with trained human or crowd coding; (2) None of the used dictionaries come close to acceptable levels of validity; and (3) machine learning, especially deep learning, substantially outperforms dictionary-based methods but falls short of human performance. From these findings, we stress the importance of always validating automatic text analysis methods before usage. Moreover, we provide a recommended step-by-step approach for (automated) text analysis projects to ensure both efficiency and validity.

**Key words**  
Sentiment Analysis; Manual Annotation; Automated Approaches; Measurement; Validity; Evaluation

Klassische Probleme (van Atteveldt et al., [2021](#): s. ähnlich Chan et al., [2021](#)):

- Schwache/keine Korrelation von Diktionären untereinander
- Negierung ignoriert
- Polysemie nicht erkannt



# Andere Verfahren: maschinelles Lernen

PA

## Automated Text Classification of News Articles: A Practical Guide

Pablo Barberá<sup>1</sup>, Amber E. Boydston<sup>2</sup>, Suzanna Linn<sup>3</sup>,  
Ryan McMahon<sup>4</sup> and Jonathan Nagler<sup>5</sup>

<sup>1</sup> Associate Professor of Political Science and International Relations, University of Southern California, Los Angeles, CA 90089, USA. Email: [pbarbera@usc.edu](mailto:pbarbera@usc.edu)

<sup>2</sup> Associate Professor of Political Science, University of California, Davis, CA 95616, USA. Email: [aboystun@ucdavis.edu](mailto:aboystun@ucdavis.edu)

<sup>3</sup> Liberal Arts Professor of Political Science, Department of Political Science, Penn State University, University Park, PA 16802, USA. Email: [sld8@psu.edu](mailto:sld8@psu.edu)

<sup>4</sup> PhD Graduate, Department of Political Science, Penn State University, University Park, PA 16802, USA (now at Google). Email: [mcmahon.rb@gmail.com](mailto:mcmahon.rb@gmail.com)

<sup>5</sup> Professor of Politics and co-Director of the Center for Social Media and Politics, New York University, New York, NY 10012, USA. Email: [jonathan.nagler@nyu.edu](mailto:jonathan.nagler@nyu.edu)

### Abstract

Automated text analysis methods have made possible the classification of large corpora of text by measures such as topic and tone. Here, we provide a guide to help researchers navigate the consequential decisions they need to make before any measure can be produced from the text. We consider, both theoretically and empirically, the effects of such choices using as a running example efforts to measure the tone of *New York Times* coverage of the economy. We show that two reasonable approaches to corpus selection yield radically different corpora and we advocate for the use of keyword searches rather than predefined subject categories provided by news archives. We demonstrate the benefits of coding using article segments instead of sentences as units of analysis. We show that, given a fixed number of codings, it is better to increase the number of unique documents coded rather than the number of coders for each document. Finally, we find that supervised machine learning algorithms outperform dictionaries on a number of criteria. Overall, we intend this guide to serve as a reminder to analysts that thoughtfulness and human validation are key to text-as-data methods, particularly in an age when it is all too easy to computationally classify texts without attending to the methodological choices therein.



Maschinelles Lernen ist oft valider (Barberá et al., [2021](#))

- Manuelle Codierung von z.B. Texten als negativ vs. positiv
- Nutzung dieser Codierungen als Trainingsdatensatz
- „Wichtige“ Features werden automatisch erkannt & für automatisierte Klassifizierung von Texten genutzt


# Noch besser: Glass-Box maschinelles Lernen

COMMUNICATION METHODS AND MEASURES  
2022, VOL. 16, NO. 4, 303–320  
<https://doi.org/10.1080/19312458.2021.1999913>

 **Routledge**  
Taylor & Francis Group

 OPEN ACCESS  Check for updates

## Enhancing Theory-Informed Dictionary Approaches with “Glass-box” Machine Learning: The Case of Integrative Complexity in Social Media Comments

Timo Dobbrick , Julia Jakob<sup>a</sup>, Chung-Hong Chan , and Hartmut Wessler 

<sup>a</sup>Mannheim Centre for European Social Research, University of Mannheim; <sup>b</sup>Institute for Media and Communication Studies, University of Mannheim

### ABSTRACT

Dictionary-based approaches to computational text analysis have been shown to perform relatively poorly, particularly when the dictionaries rely on simple bags of words, are not specified for the domain under study, and add word scores without weighting. While machine learning approaches usually perform better, they offer little insight into (a) which of the assumptions underlying dictionary approaches (bag-of-words, domain transferability, or additivity) impedes performance most, and (b) which language features drive the algorithmic classification most strongly. To fill both gaps, we offer a systematic assumption-based error analysis, using the integrative complexity of social media comments as our case in point. We show that attacking the additivity assumption offers the strongest potential for improving dictionary performance. We also propose to combine off-the-shelf dictionaries with supervised “glass box” machine learning algorithms (as opposed to the usual “black box” machine learning approaches) to classify texts and learn about the most important features for classification. This dictionary-plus-supervised-learning approach performs similarly well as classic full-text machine learning or deep learning approaches, but yields interpretable results in addition, which can inform theory development on top of enabling a valid classification.

Noch transparenter (Dobbrick et al., [2022](#)):

1. Theoretisch fundierte Identifikation von relevanten Features bzw. Diktionären
2. Diese werden als Input für Machine-Learning-Modelle genutzt

Ähnlich: semi-automatisierte „seed“-Verfahren (Watanabe, [2021](#))

# Wie berichte ich Diktions-Analysen in Papern?

*Positive/negative news.* In order to measure the amount of positive and negative news, a sentiment analysis was carried out for each article using the Sentistrength software for Dutch (Thelwall et al., 2010). Each article was assigned a score for the amount of positivity ( $M = 2.01$ ,  $SD = 1.00$ ,  $min = 1$ ,  $max = 5$ ) and negativity ( $M = -2.81$ ,  $SD = 0.90$ ,  $min = -5$ ,  $max = -1$ ) which makes it possible to compare the emotionality of different articles. As Thelwall et al. (2010) point out, sentiment is not a two-dimensional scale formed by positivity on the one and negativity on the other end: rather both are concepts that do not necessarily have to be correlated strongly and as such can (and have to be) measured individually. Therefore, by adding up the absolute values of positivity and negativity, we were able to determine emotionality ( $M = 4.82$ ,  $SD = 1.50$ ,  $min = 2$ ,  $max = 10$ ).

Wahl des  
Diktions

Analyse

Beispiel aus Burggraaf & Trilling (2020). Through a different gate: An automated content analysis of how online news and print news differ. *Journalism*.



# Wie berichte ich Diktions-Analysen in Papern?

---

- **Immer:** Relevante Schritte kurz nennen & im Appendix ausführen
  - Welches Diktionär?
  - Erfassung im gesamten Text oder Entity-spezifisch?
  - Ab wann gilt ein latentes Konstrukt als erkannt? (z. B. sobald 1 Feature aus Diktionär vorkommt?)
- **Noch besser:** Code (und ggf. Daten) teilen
- **Am besten:** Mit Multiverse-Analysen testen, wie robust Ergebnisse bei verschiedenen Diktionären bleiben, schrittweise Voraussetzungen testen (Dobbrick et al., [2022](#)) und Analysen validieren (s. **Sitzung 5!**)




# Take-Aways






- **Wichtigste Schritte:**
  - Preprocessing
  - Diktionär wählen/erstellen
  - Analyse
  - Validierung
- **Off-the-shelf Diktionäre:** Rückgriff auf bestehende Wortlisten (etwa für Emotionen, Sentiment oder Themen), oft entwickelt in anderen Kontexten
- **Organische Diktionäre:** Eigene, domänspezifische Wortlisten




# Wie geht es weiter?

## ZEITPLAN

 Mi, 24. Juli

- 09:00 - 12:00:  *Einführung & Preprocessing*
- 12:00 - 13:00:  *Mittagspause*
- 13:00 - 15:00:  *Co-Occurrence-Analysen*
- 15:00 - 17:00:  *Diktionäre*

 Do, 25. Juli

- 09:00 - 12:00:  *Topic Modeling*
- 12:00 - 13:00:  *Mittagspause*
- 13:00 - 15:00:  *Qualitätskriterien*
- 15:00 - 16:00:  *Ausblick*

# Danke! Fragen?



**Dr. Valerie Hase**  
IfKW, LMU Munich



valeriehase



valerie-hase.com



**Luisa Kutlar**  
IfKW, LMU Munich



luisakutlar