PolMine

Alle Worte sind gleich?

Term-Extraktion mit polmineR

Andreas Blaette

Stand: 19. November 2018

Zur Ermittlung der Ungleichheit des Vokabulars

- Alle Worte sind gleich?! Natürlich nicht: Manche Worte sind ungleicher als andere in dem Sinne, dass sie stärker bedeutungstragend sind als andere und den semantischen Gehalt eines Textes ausmachen. Doch wie identifiziert man das inhaltlich interessante und relevante Vokabular eines Subkorpus?

- Die bloße Häufigkeit von Worten ist bei dieser Frage ein schlechter Ratgeber. Allerweltsworte (“haben”, “machen”, “ist”, “der”, “die”, “das” etc.) treten weit häufiger auf als jenes Vokabular, das tatsächlich den Inhalt eines Textes ausmacht.

- Im ‘Text Mining’ ist es vor diesem Hintergrund gängig, Allerweltsvokabular auf Stopwort-Listen zu setzen und dieses von vornherein von der Analyse auszuschließen. Es gibt etwa eine Vielzahl von Beispielen von Wortwolken, welche die bloße Zählung der Häufigkeiten von Worten eines Textes darstellen, nachdem eine Stopwort-Liste als Filter verwendet wurde.

- Die undifferenzierte Anwendung von Stopwort-Listen im Rasenmäher-Stil kann jedoch zu ungewollten Informationsverlusten führen. Ein Beispiel: Die Pronomen “wir” und die Anrede “Sie”, die stets auf Stopwort-Listen stehen (z.B. `tm::stopwords("de")`) haben im Rollenspiel zwischen Regierungs- und Oppositionsfraktionen eine wichtige Bedeutung. Je nach Anwendungsszenario kann es schlecht begründet sein, diese Worte mit dem Rasenmäher zu eliminieren.

Statistisch signifikantes Vokabular

- In der Korpuslinguistik wurden inhaltlich besser begründete statistische Verfahren entwickelt, um das statistisch signifikante Vokabular eines (Sub-)Korpus zu ermitteln. Die Grundfrage ist dabei: Welche Worte treten in einem Untersuchtungskorpus (“corpus of interest” /) im Vergleich zu einem Referenzkorpus ( ) überzufällig oft auf?
- Diese Grundüberlegung findet in verschiedenen Domänen Anwendung
  - Diese statistischen Verfahren dienen in der Soziolinguistik der Bestimmung von “Schlagworten” bzw. . Man führt dementsprechend eine Schlagwort-Berechnung durch.
  - In Bereich des Text Mining spricht auch von einem Verfahren der “Term-Extraktion”.
  - Im Feld des Machine Learning wird auch von gesprochen: Die eines Textes machen diesen unterscheidbar von anderen.
- Die Rede von den eines Textes (oder eines Sub-Korpus) erscheint als die allgemeinste Weise, die hervorstechenden Merkmale zu bezeichnen, die in der Analyse besonders zu beachten sind. Daher ist die Methode des -Pakets, die zur Keyword-/Term-/Feature-Extraktion zur Verfügung steht, als `features()`-Methode bezeichnet.

Initialisierung

- Die Beispiele des Foliensatzes basieren auf dem -Korpus. Der Datensatz in dem -Paket wird nach dem Laden von mit der `use()`-Funktion aktiviert.

```
library(polmineR)
use("GermaParl")
```

- Weitere hier verwendete Pakete werden falls erforderlich installiert und geladen.

```
for (pkg in c("magrittr", "data.table", "RColorBrewer", "tm", "wordcloud")){
  if (!pkg %in% rownames(installed.packages())) install.packages(pkg)
  library(package = pkg, character.only = TRUE)
}
```

Die Logik des statistischen Tests

- Statistische Verfahren der Feature-Extraktion (vgl. z.B. Manning / Schuetze 2003) beruhen auf dem Vergleich des Auftretens eines Worts (oder einer lexikalischen Einheit) in einem (Sub-)Korpus ( ) im Vergleich zu einem Referenzkorpus ( ). D.h. die Zählung der Häufigkeiten in Unterschungs- und Referenzkorpus steht am Anfang des Verfahrens.

- Die zentrale Frage des an den Häufigkeiten ansetzenden Unterschiedstests ist, ob bestimmte Worte im Untersuchungskorpus oft auftreten. Diese Logik eines statistischen Unterschiedstests findet auch bei der Berechnungen von Kookkurrenzen Anwendung. Während aber bei der Kookkurrenzanalyse -Tests gängig sind, die auch bei seltener auftretenden Worten noch robust bleiben, wird bei der Feature-Extraktion oft auch mit dem etwas einfacheren -Test gearbeitet.

- Die `features()`-Methode beinhaltet voreingestellten Filter nach der Auftretenshäufigkeit, damit NutzerInnen sich über Selektionsschritte im Klaren bleiben. Es wird in aller Regel immer erforderlich sein, selten auftretendes Vokabular mit der `subset()`-Methode aus der Analyse auszuschließen.

- Die Zahl der statistisch signifikanten Worte ist in hohem Maße abhängig von der Größe des Untersuchungskorpus. Wie bei Kookkurrenz-Analysen wird die Anwendung weiterer Filter- und Reduktionskriterien oftmals relevant sein.

Ein erstes Beispiel

- Im ersten Beispiel wollen wir die herausstechenden Themen der parlamentarischen Debatte von 2015/2016 im Vergleich zu den Vorjahren ermitteln.
- Als “corpus of interest” (coi) legen wir zunächst eine `partition` für 2015/16 an, wobei wir Zwischenrufe aus der Analyse ausschließen.

```
coi <- partition("GERMAPARL", year = 2016, interjection = FALSE)
```

- Als Referenzkorpus wählen wir die vorangegangenen Jahre. Natürlich ist relevant, wie groß wir den Untersuchungszeitraum stricken. In diesem Fall wählen wir den Zeitraum ab 2002, d.h. die Phase nach dem 11. September.

```
ref <- partition("GERMAPARL", year = 2002:2015, interjection = FALSE)
```

- Nun fehlen noch die Zählungen der Worthäufigkeiten. Dies erreicht die `enrich()`-Methode.

```
coi <- enrich(coi, p_attribute = "word")
ref <- enrich(ref, p_attribute = "word")
```

Varianten des Weges zur Zählung

- Die vorangegangene Abfolge des Anlegens einer Partition mit `partition()` und der Anreicherung mit `enrich()` kann auch etwas kompakter als "Pipe" gestaltet werden:

```
coi <- partition("GERMAPARL", year = 2016, interjection = FALSE) %>% enrich(p_attribute = "word")
```

- Es ist allerdings auch möglich, das Argument `p_attribute` direkt beim Aufruf von `partition()` anzugeben und also gleich "in einem" Rutsch die Zählung durchzuführen.

```
coi <- partition("GERMAPARL", year = 2016, interjection = FALSE, p_attribute = "word")
ref <- partition("GERMAPARL", year = 2002:2015, interjection = FALSE, p_attribute = "word")
```

- Wesentlich für die Durchführung der Feature-Extraktion ist, dass Zählungen verglichen werden. Insofern ist wichtig, dass die `partition`-Objekte auch `count`-Objekte sehen. Das sehen Sie wie folgt.

```
is(coi)
```

```
## [1] "plpr_partition" "partition"      "count"          "subcorpus"
## [5] "textstat"
```

Features, endlich

- Den Vergleich besorgt die angekündigte `features()`-Methode.

```
f <- features(coi, ref)
```

- Für das `features`-Objekt können generische-Methoden wie `as.data.frame()`, `dim()`, `nrow()`, `ncol()`, `colnames()` etc. angewandt werden. Über die `nrow()`-Methode verschaffen wir uns einen Eindruck, wie viele Zeilen eigentlich die Tabelle hat.

```
nrow(f)
```

- Über 85000 Zeilen? Das ist sicher mehr als wir ansehen möchten. Also nehmen wir zwei Filterschritte vor, die stets angebracht sein werden: Wir schließen jene Worte aus, die im Untersuchungskorpus höchstens 5 mal auftreten und wir filtern (großzügig) nach dem Schwellenwert für eine Irrtumswahrscheinlichkeit von 0,1% (hier: 10,83).

```
f <- subset(f, count_coi >= 5) %>% subset(chisquare >= 10.83)
```

Das signifikante Vokabular von 2015/2016

Filter- und Reduktionsschritte

- Das berühmt-berüchtigte Rauschen in den Daten ereilt uns also auch hier. Wobei wir hier auch keine voreiligen Schlüsse ziehen sollten: Die Jahreszahlen (wie “2016”) könnten bedeuten, dass einem Jahr ein besonderer Ausnahmecharakter beigemessen wird, was inhaltlich interessant sein könnte. Jedenfalls nutzen wir eine Standard-Stopliste (aus dem -Paket) zur Bereinigung der Ergebnisse und fügen einer leicht erweiterten Liste das Auftreten von Sonderzeichen und Jahreszahlen noch hinzu.

```
terms_to_drop <- c(tm::stopwords("de"), "--", "``", "[", "]", "2016", "2017", "2015", "Vielen", "Dank")
subset(f, !word %in% terms_to_drop)
```

Filtern mit der Part-of-Speech-Annotation

- Die Ermittlung von beruht auf einem Vergleich von Häufigkeiten. Insofern eine Part-of-Speech-Annotation verfügbar ist, können wir diese als ergänzendes Merkmal in die Zählung von Worten einbeziehen. Wir schließen an die Definition der Partition via `partition()` die `count()`-Methode an und nehmen eine Zählung über zwei positionale Attribute ("word" und "pos") vor.

```
coi <- partition("GERMAPARL", year = 2016, interjection = FALSE) %>% count(p_attribute = c("word", "pos"))
ref <- partition("GERMAPARL", year = 2002:2015, interjection = FALSE) %>% count(p_attribute = c("word", "pos"))
```

- Die Berechnung der weicht zunächst nicht von dem bereits bekannten Muster ab. Den Ausschluss niedrigfrequenter Worte die Anwendung des Schwellenwerts für eine 0,1%-Irrtumswahrscheintlichkeit nehmen wir vor.

```
f <- features(coi, ref) %>% subset(count_coi >= 5) %>% subset(chisquare >= 10.83)
```

- Allerdings können wir nun auch noch nach der POS-Annotation (hier: Nomen, "NN") filtern.

```
f <- subset(f, pos == "NN")
```

Ergebnistabelle 2015 - mit POS-Filter

Vertiefung

- Wir möchten nun noch auf ein besonderes Szenario hinweisen, das relevant ist, wenn das Untersuchungskorpus Teil des Referenzkorpus ist. Dann ist es erforderlich, das Argument `included` auf `TRUE` zu setzen, so dass von der Zählung der Worte im Referenzkorpus die Häufigkeiten im Untersuchungskorpus abgezogen werden.
- Als Beispiel nehmen wir die Reden von Angela Merkel in 2008, dem Jahr der Finanzmarktkrise unter die Lupe. Wir vergleichen diese mit allen anderen Reden im Bundestag, um die Schwerpunktsetzungen der Bundeskanzlerin in Erfahrung zu bringen.

```
merkel <- partition("GERMAPARL", speaker = "Angela Merkel", year = 2008:2009, interjection = FALSE) %>%
  count(p_attribute = c("word", "pos"))
bt <- partition("GERMAPARL", year = 2008:2009, interjection = FALSE) %>%
  count(p_attribute = c("word", "pos"))
```

- Beim Aufruf der `features()`-Methode setzen wir das Argument `included` auf `TRUE`.

```
am_features <- features(merkel, bt, included = TRUE)
```

Merkel 2008

- Wir nehmen wieder die Standard-Filterschritte vor. Den Filter mit der Part-of-Speech-Annotation gestalten wir etwas weniger restriktiv als zuvor: Wir behalten als Inhaltsworte Nomen (“NN”), Adjektive (“ADJA”) und Verben (“VVFIN”).

```
am_features_min <- am_features %>%
  subset(count_coi >= 5) %>%
  subset(chisquare >= 10.83) %>%
  subset(pos %in% c("NN", "ADJA", "VVFIN"))
```

- Wortwolken sind mit Vorsicht zu genießen. Aber natürlich bietet es sich an, das extrahierte Vokabular als Wortwolke darzustellen. Die Wortwolke folgt auf der folgenden Folie, daran schließ die Ergebnistabelle an.

```
wordcloud::wordcloud(
  words = am_features_min[["word"]][1:50],
  freq = am_features_min[["count_coi"]][1:59],
  colors = rep(RColorBrewer::brewer.pal(8, "Dark2"), times = 7),
  random.color = TRUE
  )
```

Merkel 2008/2009 - eine Wortwolke

Merkel 2008/2009 - Tabelle

Mehrworteinheiten

- Die Extraktion von features beruht auf dem Vergleich von Häufigkeiten. Dabei sind es natürlich nicht nur die Häufigkeiten einzelner Worte, die verglichen werden können, sondern auch die von Mehrwort-Einheiten.
- Auch das Ergebnis einer Zählung von N-Grammen mit der `ngrams()`-Methode führt zu einem Objekt mit Zählungen, und auch hier können wir features bestimmen.

```
merkel_ngrams <- partition("GERMAPARL", speaker = "Angela Merkel", year = 2008:2009, interjection = FALSE) %>%
  polmineR::ngrams(n = 2, p_attribute = "word")
```

```
bt_ngrams <- partition("GERMAPARL", year = 2008:2009, interjection = FALSE) %>%
  polmineR::ngrams(n = 2, p_attribute = "word")
```

```
features(merkel_ngrams, bt_ngrams, included = TRUE) %>%
  subset(count_coi >= 5) %>% subset(chisquare >= 10.83)
```

Statistisch signifikante Mehrworteinheiten

Formeln und Formelhaftigkeit

- Sprache folgt standardisierten Mustern: Adjektiv-Nomen-Konstruktionen sind eine gängige Variante. Wir können die statistisch signifikanten Formen ermitteln, indem wir die Part-of-Speech-Annotation (p-Attribut “pos”) in die Zählung der N-Gramme einbeziehen.

```
merkel_ngrams <- partition("GERMAPARL", speaker = "Angela Merkel", lp = 17, interjection = FALSE) %>%
  polmineR::ngrams(n = 2, p_attribute = c("word", "pos"))

bt_ngrams <- partition("GERMAPARL", lp = 17, interjection = FALSE) %>%
  polmineR::ngrams(n = 2, p_attribute = c("word", "pos"))
```

- Wir führen die feature-Extraktion durch und wandeln das Ergebnis in ein `data.table`-Objekt um, das wir noch ein wenig formatieren.

```
dt <- features(merkel_ngrams, bt_ngrams, included = TRUE) %>% data.table::as.data.table()
dt <- subset(dt, dt[["1_pos"]] == "ADJA") %>% subset(.[["2_pos"]] == "NN")
dt[,"1_pos" := NULL][,"2_pos" := NULL][,"exp_coi" := round(exp_coi, 2)][, "chisquare" := round(chisquare, 2)]
```

- Die Darstellung erfolgt auf der folgenden Folie.

Adjektiv-Nomen-Konstruktionen

Nomen-Artikel-Nomen-Konstruktionen

- Ein letztes Beispiel, das demonstrieren soll, dass die Analyse umfassenden Entfaltungsspielraum findet: Wir können zu Mehrwort-Formeln aus drei Worten übergehen und Nomen-Artikel-Nomen-Konstruktionen ermitteln. Beim ersten Schritt erhöhen wir lediglich die Zahl der analysierten Worte.

```
cdu_ngrams <- partition("GERMAPARL", party = "CDU", lp = 17, interjection = FALSE) %>%
  polmineR::ngrams(n = 3, p_attribute = c("word", "pos"))

non_cdu_ngrams <- partition("GERMAPARL", party = c("SPD", "FDP", "LINKE", "GRUENE"), lp = 17, interjection = FALSE)
  polmineR::ngrams(n = 3, p_attribute = c("word", "pos"))

cdu_ngrams <- subset(cdu_ngrams, count >= 5) # vorgezogen

f <- features(cdu_ngrams, non_cdu_ngrams, included = FALSE)
f <- subset(f, chisquare >= 10.83)
```

- Wir wandeln das in ein `data.table`-Objekt um und machen uns ans Filtern.

```
dt <-  data.table::as.data.table(f)
dt <- subset(dt, dt[["1_pos"]] == "NN") %>% subset(.[["2_pos"]] == "ART") %>% subset(.[["3_pos"]] == "NN")
```

Nomen-Artikel-Nomen-Konstruktionen

Ausblick

- Verfahren der feature-Extraktion, die im polmineR-Paket mit der `features()`-Methode implementiert sind, können für eine Reihe von Zwecken eingesetzt werden:
- Als Verfahren zur Schlagwort-Berechnung kann effizient das semantisch tragende Vokabular extrahiert werden, so dass effizient zentrale Begriffe des Diskurses erkannt werden können.
- Wenn man dem Vergleich ein thematisch definiertes Subkorpus zugrunde legt, lässt sich das semantische Feld des Themenbereichs bestimmen. Dies kann etwa der Ansatzpunkt für die Entwicklung von Diktionären sein.
- Die Ermittlung statistisch signifikanten Vokabular kann auch für Zwecke der Bestimmung von features genutzt werden, auf die Zählungen oder andere textstatistische Maße eingeschränkt werden, so dass Speicherplatz geschont und die Qualität von Ergebnissen verbessert wird.
- Insgesamt sind die geschilderten Verfahren ein flexibles wie effizientes Instrument, um effizient in Sachen Vokabular Wichtiges von Unwichtigem zu unterscheiden.