Text as Linguistic Data: Korpusanalyse mit polmineR

Andreas Blaette

16 April 2018

Warum Korpusanalyse mit R?

- R als sozialwissenschaftliche ‘lingua franca’
- Interaktivität
- freundliche Nutzercommunity
- RStudio als IDE
- dynamische Weiterentwicklung von R, RStudio

Zielsetzungen von polmineR

- Interaktivität
- Validität
- Performanz
- Quelloffenheit (Open Source)
- Portabilität (Nutzung unter Windows, macOS, Linux)
- Nutzerfreundlichkeit
- Dokumentation
- Reproduzierbarkeit

Getting started

Installationsmöglichkeiten

- Linux, MacOS
- virtualisiertes Unix unter Windows
- Serverinstallation, Nutzung von RStudio Server

Systemvoraussetzungen

- (möglichst) mehr als 4GB Arbeitsspeicher
- mehrere Prozessorkerne
- SSD

Installation

** Dependencies des polmineR-Pakets: **

- methods (S4-Klassensystem)
- rcqp (Zugriff auf die CWB)
- slam (simple triplet matrix)
- Matrix (Matrizen)

Erste Schritte

** Initialisierung **

```
library(polmineR)
use("bt")
```

** Anlegen einer Partition **

```
regerklaerung <- partition(
  "BT",
  list(speaker_name="Angela Merkel", speaker_date="2005-11-
  type="plpr"
  )
```

** Lesen **

```
read(regerklaerung, meta="speaker_name")
read(regerklaerung, meta=c("speaker_name", "speaker_date",
```

** Konkordanzen **

```
bt2002 <- partition("BT", list(plenary_protocol_year="2002"
```

Anlegen von Partition

Eine wichtige Unterscheidung

- s-Attribute (parameter sAttributes)
- p-Attribute (parameter pAttributes)

Exploration eines Korpus

```
pAttributes("BT")
```

```
sAttributes("BT")
sAttributes("BT", "speaker_name")
```

. . . aber das kann man ohne Ursprungsdaten nicht verstehen:

Kurzeinführung in XML (vgl. HTML)

- Hierarchie
- Wohlgeformtheit
- (DTD-)Validität

Frequenzzählungen

count()

```
btByYears <- partitionBundle(
  "BT",
  def=list(speaker_type="speech"),
  var=list(speaker_year=NULL)
  )
islam <- count(
  btByYears,
  query=c('Islam', 'Muslime', 'Terror'),
  pAttribute="word", mc=FALSE
  )
islam2 <- as.data.frame(islam)[, c(2:ncol(islam))]
rownames(islam2) <- islam[["partition"]]
library(bubblegraph)
linechart(as.data.frame(t(islam2)))
```

```
mig <- count(
  btByYears,
```

Frequenzzählung mit CQP-Syntax

** Nutzung der CQP-Syntax **

Grundlagen: - Ansteuern von p-Attributen - Quantoren - Platzhalter

```
sttsTagsetInfo <- "http://www.ims.uni-stuttgart.de/forschun
browseURL(sttsTagsetInfo)
```

http://cwb.sourceforge.net/files/CQP_Tutorial/
http://cwb.sourceforge.net/files/CQP_Tutorial.pdf
http://www.ims.uni-stuttgart.de/forschung/projekte/CorpusWorkbench/CQPTutorial/cqp-tutorial.2up.pdf

```
btByYear <- partitionBundle(
  "BT",
  def=list(speaker_type="speech"),
  var=list(speaker_year=as.character(c(1998:2007)))
  )

mig <- count(
  btByYear,
```

Konkordanzen

```
kwic(bt, '"Krieg" "gegen" "den" "Terror"', meta=c("speaker_
kwic(bt, '"Krieg" "gegen" "den" "Terror"', meta=c("speaker_
```

```
foo <- frequencies(bt, '"()"')

Bkwic(bt, '"Krieg" "gegen" []{0,1} [pos="NN"]', meta=c("spe
```

Inspektion der Hilfe ...

Und ein Beispiel: Prävention in der Sozialpolitik

Kollokationsanalysen

```
bt <- partition("BT", list(speaker_year="2006"), regex=TRU
islam <- context(bt, "Islam", pAttribute="word")
islam <- context(bt, '"Islam.*"', pAttribute=c("word", "pos
view(islam)
islam2 <- subset(islam, pos %in% c("NN", "ADJA"))
view(islam2)
wordcloud(
  words=islam2@stat[["word"]][1:50],
  freq=islam2@stat[["ll"]][1:50]/2,
  colors=brewer.pal(8,"Dark2")
  )
dotplot(islam2, col="ll", 25)
```

weiterführend: Als Testverfahren implementiert t-test, PMI, log-likelihood

** Beachte: ** - Abhängigkeit der statistischen Testwerte von der Korpusgröße - keine unmittelbare Vergleichbarkeit der Testwerte! - qualitative Validierung

Schlagwortanalysen

Was nicht funktioniert ...

```
schroeder1 <- partition("BT",
  list(speaker_name="Gerhard Schröder", speaker_date="2001-
  type="plpr"
)
schroeder2 <- partition("BT",
  list(speaker_name="Gerhard Schröder", speaker_date="2001-
  type="plpr"
)
schroeder3 <- partition("BT",
  list(speaker_name="Gerhard Schröder", speaker_date="2001-
  type="plpr"
)

bt2001 <- partition("BT", list(speaker_year="2001"), pAttr

keyws1 <- compare(schroeder1, bt2001)
keyws2 <- compare(schroeder2, bt2001)
```

Perspektiven der Textstatistik

```
merkel <- partition("BT", list(speaker_name=".*Merkel.*", s
merkelSpeeches <- as.speeches(
  merkel, sAttributeDates="speaker_date", sAttributeNames="
  gap=500
  )
merkelSpeeches <- enrich(merkelSpeeches, pAttribute="word")
dtm <- as.DocumentTermMatrix(merkelSpeeches, col="count")
toDrop <- polmineR:::noise(dtm)
dtmTrimmed <- trim(dtm, termsToDrop=unique(unlist(toDrop)))
dtmTrimmed <- trim(dtmTrimmed, docsToDrop = names(which(sla

library(topicmodels)
tmodel <- LDA(
  dtmTrimmed, k=20, method = "Gibbs",
  control = list(burnin = 1000, iter = 50, keep = 50, verbo
)
View(terms(tmodel, k=20))
```

Ausblick: Korpusaufbereitung

```
xmlify(bt, sourceDir="txt_utf8", targetDir="tei", pattern="

# the following steps may have to be repeated until the da
# - match a key generated from the speaker attributes agai
# - add the information to the TEI documents
# - inspect whether there is still information missing
# - pimp the alias file, repair wikipedia data etc

partyAffiliationOfSpeakers <- getPartyAffiliation(bt, sourc
addSpeakerAttributes(bt, sourceDir="tei", targetDir="tei_en
missingInfo <- getMissingInformation(bt, sourceDir="tei_enr

xslt(bt, sourceDir="tei_enriched", targetDir="cwbxml", verb
tokenize(bt, sourceDir="cwbxml", targetDir="tok", with="tre
treetagger(bt, sourceDir="tok", targetDir="vrt", progress=T
fixVrt(bt, sourceDir="vrt", targetDir="vrt_fixed", mc=9, ve
adjustEncoding(bt, sourceDir="vrt_fixed", targetDir="vrt_la
cwbImport(bt, sourceDir="vrt_latin", "PLPRBTTEI", xml=TRUE,
```