Text Mining in R
(**tm** 101)

ViennaR

Mario Annau, 22.2.2016

Textmining?

- Statistical analysis of textual data
- Use Cases include
  - Spam Filtering
  - Search
  - Sentiment Analysis
  - Topic Modelling
  - ...

tm?

- Infrastructure to Analyze Collections of Texts (Corpora) in R
- Typical **tm** pipeline:
  1. Read Data from Numerous Sources into Corpus
  2. Preprocess Data
  3. Create DTM/TDM
  4. Apply Model

Contents

- Data Reading
- Data Structures
- Preprocessing Pipeline

  removePunctuation, tolower, removeWords, stripWhitespace, stemDocument
- Examples
- Known Weaknesses and Outlook
- Plans for **SentimentAnalysis** (tm.plugin.sentiment 2.0)

Data Reading

- **tm** Separates Data *Source* and *Reader (Iterator)*
- Supported Data Sources and Readers:

```
R> tm::getSources()
[1] "DataframeSource" "DirSource"       "URISource"
"VectorSource"
[5] "XMLSource"       "ZipSource"
R> tm::getReaders()
 [1] "readDOC"                "readPDF"
 [3] "readPlain"              "readRCV1"
 [5] "readRCV1asPlain"        "readReut21578XML"
 [7] "readReut21578XMLasPlain" "readTabular"
 [9] "readTagged"             "readXML"
```

- e.g. Read PDF Files from Directory:

```
R> Corpus(DirSource(directory = ".", pattern = "*.pdf"),
readerControl = list(reader = readPDF, language = "en"))
```

Data Structures

- TextDocument (**NLP**)
- Annotations (**NLP**)
- Corpus
- DocumentTermMatrix

Preprocessing Pipeline

```
R> removePunctuation("This is awesome and
cool!")
[1] "This is awesome and cool"
R> tolower("This is awesome and cool!")
[1] "this is awesome and cool!"
R> removeWords("This is awesome and cool!",
stopwords())
[1] "This  awesome  cool!“
R>stripWhitespace(removeWords(tolower(removePunc
tuation("This is awesome and cool!")),
stopwords()))
[1] " awesome cool“
R>stemDocument(crude[[1]])
```

Document Term Matrix

```
R> control = list(
  removePunctuation = TRUE,
  removeNumbers = TRUE,
  tolower = TRUE,
  removeWords = list(stopwords("english")),
  stripWhitespace = TRUE,
  stemDocument = TRUE)
R> dtm <- DocumentTermMatrix(crude, control=control)
```

Calculate Simple Sentiment Score

- We can now use the DTM to calculate sentiment scores based on dictionary
- e.g.

```
sentiment <- DocumentTermMatrix(crude,
control=control)
pos <- tm_term_score(dtm, dic_gi$positive, FUN =
slam::row_sums)
neg <- tm_term_score(dtm, dic_gi$negative, FUN =
slam::row_sums)
sentiment <- (pos - neg) / (pos + neg)
```

Known Weaknesses

- ?

SentimentAnalysis package

- **tm, tm.plugin.sentiment** -> bag of words approach with caveats
- **syuzhet** -> nice collection of techniques, quite different goals
- **coreNLP**
- Datasets? -> Bing Liu