Text Mining

Fundamentals of Computing and Data Display

Christoph Kern   Ruben Bach[1]

(c.kern, r.bach)@uni-mannheim.de

[1]Based on Klochikhin & Boyd-Graber, 2020 and Welbers et al. 2017

Outline

1. Text Data in the Social Sciences
2. Typical steps
3. Resources

Text as data

- Long tradition in the social sciences
  - Content analysis (communication studies, political science, sociology...)
  - Open-ended survey questions
- With the rise of the internet, tons of new data sources
  - Social media data
  - "Internet" data
  - Automatic transcripts of speeches, videos, news, ...
- Requires new analytical techniques!
- Computational text analysis (CTA), quantitative text analysis, text mining, ...

Text as data

New challenges

- Social scientists used to work with structured data (e.g., survey data)
- Text often comes as unstructured data (characters, words, sentences, paragraphs, ...)
- Text and language often many nuances, ambiguous meaning, sarcasm, ...
- CTA often
  - Requires a lot of (simplifying) assumptions – e.g. standard English (Social media?!)
  - Is more qualitative/subjective than the methods suggests

Text as data - Examples

Method	**Goal**	**Examples**
Search	Finding relevant content	Literature reviews
Topic detection and Clustering	Understanding what text is about	Understanding social media content
Classification	Classifying text into predefined categories	Detecting trolls and bots in social networks; Identifying fake news
Sentiment analysis	Understanding sentiment or polarity of a text	What do social media users think of politicians
Word clustering/Synonyms	Finding words that transport similar meaning	Understanding social media content
Named entity recognition	Recognition, tagging and extraction of named entities	Automated analysis of laws, court rulings, etc.
General extraction	Recognition, tagging, and extraction of specific classes of words	Understanding user activities from social media
Visualization	Visualization of text data	Networks of politicians
Summarization	Automated summarization of long texts	Laws, news media content, diaries
Translation	Automating translation from one language	Understanding social media and news content
Translation	to another	across countries

Text as data - Methods overview

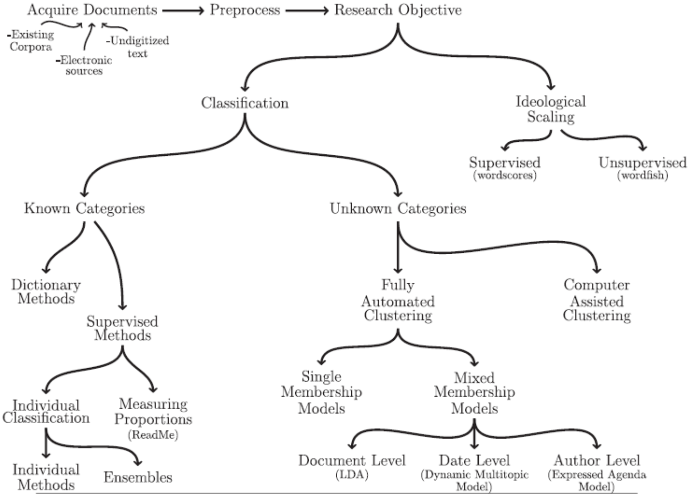

Fig. 1 An overview of text as data methods.

Figure: Grimmer and Stewart, 2013

Text as data - Typical steps

Text data requires a lot of pre-processing

- Initial Processing: Get raw text, remove unnecessary content. Split up sentences in words, remove unnecessary words.
- (Adding Linguistic Features: Part-of-speech tags – identify grammatical structure)
- Converting text to a (sparse) matrix: Define rows, columns and cell content
- Analysis

Pre-processing

Cleaning and processing text

- Tokenization
  - Text: "X and Y are 2 Kremlin trolls! Trolling day and night for a few rubles."
  - Sentences: ["X and Y are 2 Kremlin trolls !","Trolling day and night for a few rubles."]
  - Words / Unigrams ["X", "and", "Y", "are", "2", "Kremlin", "trolls", "!","Trolling", . . . , "rubles","."]
  - Letters
  - N-grams (Unigrams, Bigrams, Trigrams, ...)
  - Skip-grams: 1-skip2-grams: ["X Y", "and are", "Y 2", "are Kremlin", ..., "a rubles"]

Pre-processing

Cleaning and processing text

- Stopwords
  - Remove words that transport little semantic meaning: prepositions, articles, common nouns, etc
  - ”and”, ”are”, ”also”, ...
  - ![”and”,”a”,”for”,”few”]
  - However, sometimes we do need them for our analysis!
- Remove capitalization

Pre-processing

Cleaning and processing text

- Stemming and Lemmatization
  - Reducing inflected words to their word stem
  - Cutting off common suffixes
    - trolling - troll
    - trolls - troll
    - rubles - rubl
    - systems - system, systematic - system, systemic - system
- Lemmatization: based on morphological analysis of each word (are - be, go / went/ goes / gone - go)

Linguistic Analysis

Part-of-speech tagging

- Incorporate meaning of word and the way it is used in analysis
- Allows better understanding of text: verb? noun? prep? adj? adv?
- Various techniques: Rule-based, stochastic, ...
- Position matters: "A plants/N needs light and water." –"Each one plant/V one."
- Remove stopwords?

Turning Text into a Matrix

- Processing results in columns
- Rows: Sentences, words? "Tokens"
- Cell: Binary indicator, count, ...?
- Weighting often included as many words will occur very often with little added information
  - How often does a word occur in one document compared to the overall collection of docs?
  - **TF-IDF**: Value increases proportionally to number of times a word appears in a document. Offset by number of documents in corpus that contain the word. Adjust for the fact that some words appear more frequently in general.

Analysis

Topic modeling

- Topics are probability distributions over words
- Most popular method: Latent Dirichlet Allocation (LDA)
- Key idea
  - Topics form building blocks of a corpus
  - Topics are distributions over words
  - Often shown as probability-ranked list of words
  - Do not know (numberof) topics a priori
  - Goal is to discover (number of) topics
- Each document in a corpus can be explained by a number of topics: each document has an allocation over latent topics governed by a Dirichlet distribution
- Could then use topics inferred from text to classify new documents

Analysis

Topic modeling

Table: Key terms by topic

Political news	Sports news	Int. news	Econ. news
percent	game	UN	dollar
poli	sport	China	stock
party	match	Europe	tax
trump	coach	NATO	gross
GOP	league	treat	million

Analysis

Sentiment Analysis

- Humans use understanding of emotional intent of words to infer whether a section of text is positive or negative
- Sentiment analysis allows us to automate this task
- One (simple) approach
  - Consider a text as a combination of its individual words
  - Sentiment content of whole text is sum of the sentiment content of the individual words
  - Sentiments of words provided as a dictionary of words with associated sentiments
    - SentiWord: more than 150k words, each with a score between -1 and 1
    - Vader: Specialized dictionary for social media texts
    - NRC Word-Emotion Association Lexicon (EmoLex): associates words with 10 sentiments: positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust

Analysis

Advanced Natural Language Processing (NLP)

- Represent words as vectors that machines can understand
- Similar words should be close in vector space ("result in similar vectors") (e.g. dog / poodle)
- word2vec (Mikolov et al. 2013): predicts, given a word, whether another word will appear in same context
- BERT (Devlin et al. 2019) and ELMO (Peters et al. 2018) are state-of-the-art deep learning approaches for representing words in vectors
  - Also consider context of a word within a sentence
  - Difficult to understand, require quite some (Python) programming skills

Resources

Based on Welbers et al. (2017)

	R packages	
Operation	example	alternatives
Data preparation		
importing text	*readtext*	*jsonlite, XML, antiword, readxl, pdftools*
string operations	*stringi*	*stringr*
preprocessing	*quanteda*	*stringi, tokenizers, snowballC, tm, etc.*
document-term matrix (DTM)	*quanteda*	*tm, tidytext, Matrix*
filtering and weighting	*quanteda*	*tm, tidytext, Matrix*
Analysis		
dictionary	*quanteda*	*tm, tidytext, koRpus, corpustools*
supervised machine learning	*quanteda*	*RTextTools, kerasR, austin*
unsupervised machine learning	*topicmodels*	*quanteda, stm, austin, text2vec*
text statistics	*quanteda*	*koRpus, corpustools, textreuse*
Advanced topics		
advanced NLP	*spacyr*	*coreNLP, cleanNLP, koRpus*
word positions and syntax	*corpustools*	*quanteda, tidytext, koRpus*

Resources

```r
install.packages("quanteda")
library(quanteda)

text <- "An example of preprocessing techniques"
toks <- tokens(text)  ## tokenize into unigrams
toks
```

```
tokens from 1 document.
text1 :
[1] "An"  "example"  "of"  "preprocessing"  "techniques"
```

Resources

```r
sw <- stopwords("english")    ## get character vector of stopwords
head(sw)                      ## show head (first 6) stopwords
```

```
[1] "i"      "me"     "my"     "myself" "we"     "our"
```

```r
tokens_remove(toks, sw)
```

```
text1 :
[1] "exampl"     "preprocess"     "techniqu"
```

Resources

```
toks <- tokens_tolower(toks)
toks <- tokens_wordstem(toks)
toks
```

```
[1] "an"     "exampl"     "of"     "preprocess"     "techniqu"
```

Resources

```r
toks <- tokens_tolower(toks)
toks <- tokens_wordstem(toks)
toks
```

```
[1] "an"     "exampl"     "of"     "preprocess"     "techniqu"
```

Resources

```
text <-  c(d1 = "An example of preprocessing techniques",
           d2 = "An additional example",
           d3 = "A third example")
dtm <- dfm(text,                                ## input text
           tolower = TRUE, stem = TRUE,         ## set lowercasing and stemming to TRUE
           remove = stopwords("english"))       ## provide the stopwords for deletion
dtm
```

```
Document-feature matrix of: 3 documents, 5 features (53.3\% sparse).
3 x 5 sparse Matrix of class "dfmSparse"
          features
docs         exampl preprocess techniqu   addit  third

  d1              1          1        1       0      0
  d2              1          0        0       1      0
  d3              1          0        0       0      1
```

Resources

```
fulltext <- corpus(rt)                                    ## create quanteda corpus
dtm <- dfm(fulltext, tolower = TRUE, stem = TRUE,         ## create dtm with preprocessing
           remove_punct = TRUE,remove = stopwords("english"))
dtm
```

```
Document-feature matrix of: 5 documents, 1,405 features (67.9% sparse).
```

Resources

```
install.packages("topicmodels")
library(topicmodels)

texts = corpus_reshape(data_corpus_inaugural, to = "paragraphs")

par_dtm <- dfm(texts, stem = TRUE,              ## create a document-term matrix
               remove_punct = TRUE, remove = stopwords("english"))
par_dtm <- dfm_trim(par_dtm, min_count = 5)     ## remove rare terms
par_dtm <- convert(par_dtm, to = "topicmodels") ## convert to topicmodels format

set.seed(1)
lda_model <- topicmodels::LDA(par_dtm, method = "Gibbs", k = 5)
terms(lda_model, 5)
```

```
         Topic 1       Topic 2     Topic 3      Topic 4       Topic 5
[1,]     "govern"      "nation"    "great"      "us"          "shall"
[2,]     "state"       "can"       "war"        "world"       "citizen"
[3,]     "power"       "must"      "secur"      "new"         "peopl"
[4,] "constitut"       "peopl"     "countri"    "american"    "duti"
[5,]       "law"       "everi"     "unit"       "america"     "countri"
```

Resources

Sentiment analysis in R

- **Vader** – especially helpful for social media texts: `https://cran.r-project.org/web/packages/vader/index.html`
- **syuzhet**: `https://cran.r-project.org/web/packages/syuzhet/index.html` comes with AFINN Bing NRC lexicons (aka EmoLex)
- **sentimentR** - `https://github.com/trinker/sentimentr` Takes into account valence shifters (i.e., negators, amplifiers (intensifiers), de-amplifiers (downtoners), and adversative conjunctions) while maintaining speed
- **tidytext** – comes with AFINN, Bing and NRC lexicons/EmoLex (use "sentiment" dataset that comes with tidytext package). See `tidytextmining.com` for application
- **SentimentAnalysis** - `https://www.rdocumentation.org/packages/SentimentAnalysis/versions/1.3-3` comes with additional lexicons (e.g., Harvard IV, finance-specific lexicons)
- **Sentiword**: `https://github.com/aesuli/SentiWordNet` very large lexicon

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Klochikhin, E. and Boyd-Graber, J. (2020). Text Analysis. In Foster, I., Ghani, R., Jarmin, R. S., Kreuter, F., and Lane, J. (Eds.). *Big Data and Social Science: A Practical Guide to Methods and Tools*. Boca Raton, FL: CRC Press Taylor & Francis Group.

Grimmer, J. and Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, 21(3), 267–297.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013.) Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems*, 3111–9. Morgan Kaufmann.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C. , Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Welbers, K., Van Atteveldt, W., and Benoit, K. (2017). Text Analysis in R. *Communication Methods and Measures*, 11(4), 245–265.