Text as Data

Zoltan Fazekas
zfazekas.github.io

19 November 2015
@cph ssd

Word clouds are the pie charts of text analysis!

Today

A satelite look:

some basic principles (1), different goals & methods (2), and example (3)

Resources

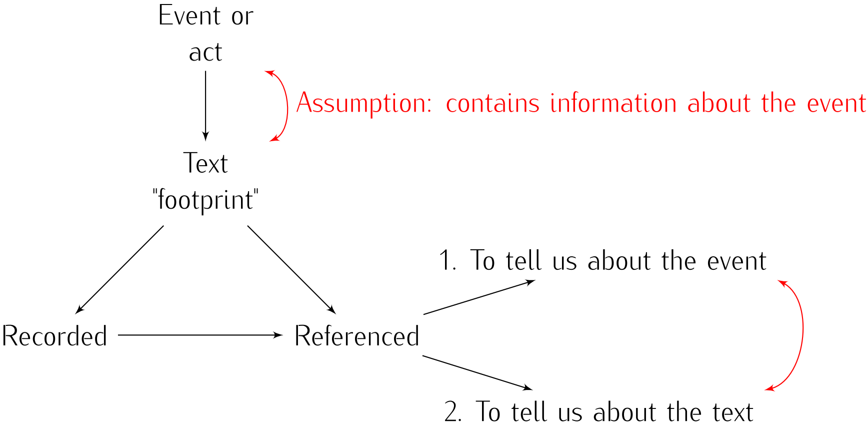
- Names (selective)
  - Will Lowe, Justin Grimmer, Kenneth Benoit, Margaret E. Roberts, Sven-Oliver Proksch
- R packages
  - **`tm, austin, quanteda, stm, RTextTools, stringr`**
- No matter how frustrating: regular expressions

Some goals

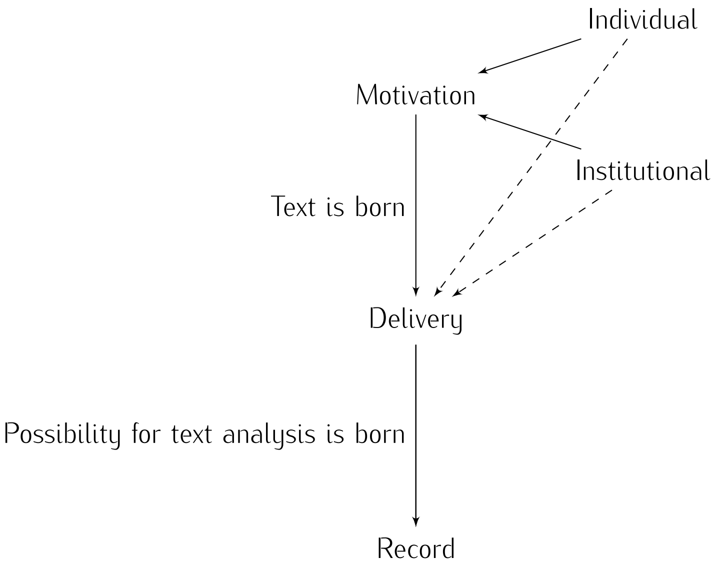
1. **Reveal** mechanisms according to which words influence and are influenced by human behavior (Roberts, 2000)
2. **Systematic** analysis of large scale text collections (Grimmer & Stewart, 2013)

We want to understand society (or the social) as expressed through words, but should this understanding be based on our conception (theory) of society or simply identify the intended meaning?

General framework



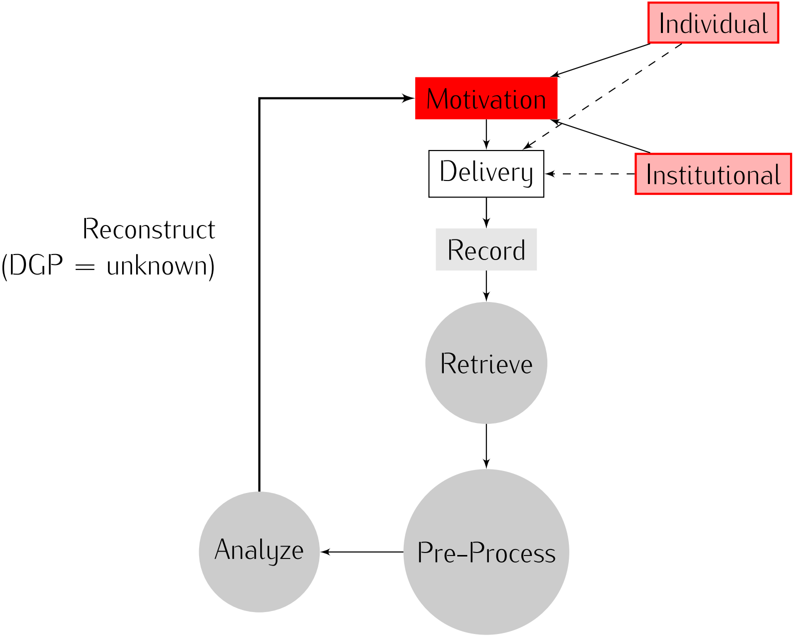
General framework



From text to data

- Requirements
  - Transform to *something* that can serve as input for analysis
  - What makes texts similar or different?
- Word (token) frequency, shared and unshared tokens – term-document matrix/document-term matrix
- (common) Assumption: bag of words
- Uni-grams, bi-grams, *n-grams*
- All tokens supposedly informative?
  1. Pre-processing – which steps and why?
  2. Substantive decisions

The grand scheme



The wide variety

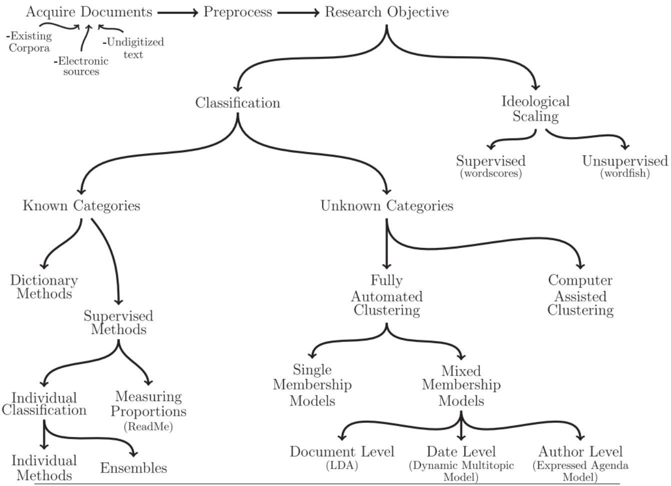


Fig. 1 An overview of text as data methods.

For some branches

The 'one' way

Lowe, W. (2013). There's (basically) only one way to do: Some unifying theory for text scaling models.

No matter what: VALIDATE!

Application

Example: which texts?

- Prime minister's opening addresses, Denmark, 1953-2013
- (substantive) Properties you might want to consider:
  - When?
  - Where?
  - Why?

Example: content

- An account of the current state of Danish affairs (established in the § 38 (1) of the Danish Constitutional Act): (1) overarching, (2) mixture of 'what has been done' and 'what will be done'
- Touches upon multiple domains, or 'topics'
- Given the current state of Danish affairs and government priorities:
  - Some topics are selected to be included (limited space)
  - Some topics are addressed more in detail
- Non-technical political speech (with extended general interest in recent years, i.e. broadcast)

Example: metadata and tasks

- Year, prime minister (who gave the talk), prime minister's party, coalition government – single party government
- Goals
    1. Load, inspect, and pre-process texts
    2. Classification/prediction application: elections next year?

Before we start

THE NEW YORKER

"I think whatever's going to happen next has already happened."

Example: follow along

- Code: https://zfazekas.github.io/resources/text_class/text_classification.R

```
data_path <- "https://zfazekas.github.io/resources/text_class/data.zip"
download(data_path, dest = "data.zip", mode = "wb")
unzip("data.zip", exdir = "./")
```

Example: some metadata

```
library("dplyr")
pm <- read.table("./data/pm-data.txt", sep = "\t",
                 header = TRUE,
                 stringsAsFactors = FALSE, encoding = "UTF-8")
elections <- read.csv("./data/elections.csv", header = TRUE,
                      stringsAsFactors = FALSE)
head(elections)
```

```
##   year next_elect
## 1 1953  5/14/1957
## 2 1954  5/14/1957
## 3 1955  5/14/1957
## 4 1956  5/14/1957
## 5 1957 11/15/1960
## 6 1958 11/15/1960
```

Example: some metadata

```
elections$next_date <- as.Date(elections$next_elect,
                               format = "%m/%d/%Y")
elections$speech <- as.Date(paste0("10/3/", elections$year),
                            format = "%m/%d/%Y")
elections$dist_weeks <- difftime(elections$next_date,
                                 elections$speech,
                                 unit = "weeks") %>%
                        round(., 0) %>%
                        as.numeric(.)
elections$dist_category <- "0"
elections$dist_category[elections$dist_weeks < 51] <- "1"
pm <- merge(pm, elections[, c("year", "dist_category")],
            by = "year")
```

Texts

```r
library("tm")
tm_corp <- Corpus(DirSource("./data/pm_speeches"),
                         readerControl = list(language = "da"))
pm$texts <- sapply(tm_corp, function (x) paste(x, collapse = " "))

library("quanteda")
pm_corp <- corpus(pm$texts, docvars = pm[, 1:5])
```

Collecting specifics: PM names

```r
library("stringr")
library("dplyr")

pm_name <- docvars(pm_corp)$pm %>%
  unique(.) %>% tolower() %>%
  paste(., collapse = " ") %>%
  str_split(., " ") %>%
  unlist() %>%
  unique(.)

pm_name
```

```
##  [1] "hans"           "hedtoft"        "christian"
##  [4] "hansen"         "viggo"          "kampmann"
##  [7] "jens"           "otto"           "krag"
## [10] "hilmar"         "baunsgaard"     "anker"
## [13] "jørgensen"      "poul"           "hartling"
## [16] "schlüter"       "nyrup"          "rasmussen"
## [19] "anders"         "fogh"           "lars"
## [22] "løkke"          "helle"          "thorning-schmidt"
```

Corpus

```r
tail(summary(pm_corp, verbose = FALSE))[, 1:7]
```

```
## Corpus consisting of 61 documents.

##          Text Types Tokens Sentences year party coalition
## text56 text56  1401   4799       449 2008     V         1
## text57 text57  1498   5114       391 2009     V         1
## text58 text58  1342   4649       412 2010     V         1
## text59 text59  1338   4946       497 2011     S         1
## text60 text60  1246   4442       424 2012     S         1
## text61 text61  1373   4871       424 2013     S         1
```

Document-feature matrix

```r
pm_dfm <- dfm(pm_corp, language = "danish",
              toLower = TRUE,
              removePunc = TRUE,
              removeSeparators = TRUE,
              stem = TRUE
              )
```

```
## Creating a dfm from a corpus ...
##    ... lowercasing
##    ... tokenizing
##    ... indexing documents: 61 documents
##    ... indexing features: 20,688 feature types
##    ... stemming features (Danish), trimmed 7214 feature variants
##    ... created a 61 x 13474 sparse dfm
##    ... complete.
## Elapsed time: 1.238 seconds.
```

Document-feature matrix

```r
head(pm_dfm)
```

```
## Document-feature matrix of: 61 documents, 13,474 features.
## (showing first 6 documents and first 6 features)
##        features
## docs    der majestæt æred medlem  af folketing
##   text1  59        3    1      2  93         5
##   text2  61        0    0      1  98         7
##   text3  74        0    0      0 113         4
##   text4  65        0    0      1 115         6
##   text5  69        0    0      1 123         2
##   text6  63        0    0      0 122         3
```

Additional terms

```
folk_terms <- grep("folket", colnames(pm_dfm), value = TRUE)
dk_terms   <- grep("dansk", colnames(pm_dfm), value = TRUE)
rem_terms  <- c("ing", "ning", "vor", "fordi", "danmark",
               "vores", "derfor", "mellem", "mere", "tak",
               "ingen", "majestæt", "kong", "dronning",
               dk_terms, folk_terms, pm_name)
length(rem_terms)
```

```
## [1] 74
```

Stopwords and collected features

```
pm_dfm <- dfm(pm_corp, language = "danish",
              toLower = TRUE,
              removePunc = TRUE,
              removeSeparators = TRUE,
              stem = TRUE,
              ignoredFeatures = c(stopwords("danish"),
                                  rem_terms),
              verbose = FALSE
              )
```

```
head(pm_dfm)
```

```
## Document-feature matrix of: 61 documents, 13,351 features.
## (showing first 6 documents and first 6 features)
##        features
## docs    æred medlem bring ærbød overvær først
##   text1    1      2     3     1       1     4
##   text2    0      1     0     0       0     4
##   text3    0      0     1     0       0     8
##   text4    0      1     1     0       0     6
##   text5    0      1     1     0       0     3
```

Trimming

```r
pm_dfm <- trim(pm_dfm, minDoc = 9) ## 15% of documents
```

```
## Features occurring in fewer than 9 documents: 11526
```

```r
dim(pm_dfm)
```

```
## [1]   61 1825
```

```r
head(pm_dfm)
```

```
## Document-feature matrix of: 61 documents, 1,825 features.
## (showing first 6 documents and first 6 features)
##        features
## docs     regering ikk kan bliv vær år
##   text1        38   8   9   18   7  4
##   text2        43  16  12   26  16 16
##   text3        34  14   6   33  15 23
##   text4        33   3  10   31  17 20
##   text5        46   6   9   45  14 22
##   text6        39   8  13   50  16 18
```

CLassification

```
total <- 1:61 ## total # documents
set.seed(162648)
train_docs <- sample(1:61, 40, replace = FALSE) ## training set
test_docs  <- total[total %in% train_docs == FALSE] ## test set
library("RTextTools")
pm_cont <- create_container(pm_dfm,
                            docvars(pm_corp)$dist_category,
                            trainSize = train_docs,
                            testSize = test_docs,
                            virgin = FALSE)
## Train
support_train <- train_model(pm_cont, "SVM")
glm_train <- train_model(pm_cont, "GLMNET")

## Classify
support_class <- classify_model(pm_cont, support_train)
glm_class <- classify_model(pm_cont, glm_train)
```

How did we do?

```
analytics <- create_analytics(pm_cont,
                              cbind(support_class, glm_class))
summary(analytics)
```

```
## ENSEMBLE SUMMARY
##
##        n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL
## n >= 1                1.00              0.57
## n >= 2                0.76              0.69
##
##
## ALGORITHM PERFORMANCE
##
##    SVM_PRECISION        SVM_RECALL        SVM_FSCORE GLMNET_PRECISION
##            0.590             0.535             0.505            0.500
##    GLMNET_RECALL     GLMNET_FSCORE
##            0.500             0.475
```

```
doc_summary <- analytics@document_summary
```

Housekeeping

```
svm_results <- paste0(doc_summary[, 1],
                      " (",
                      round(doc_summary[, 2], 3),
                      ")")
glm_results <- paste0(doc_summary[, 3],
                      " (",
                      round(doc_summary[, 4], 3),
                      ")")
all <- data.frame(docvars(pm_corp)[test_docs, c(1, 3, 5)],
                  svm_results, glm_results)
```

Where did we do well?

```r
all[1:10, ]
```

```
##        year coalition dist_category svm_results glm_results
## text1  1953         0             0   0 (0.724)   0 (0.787)
## text5  1957         1             0   1 (0.614)   0 (0.882)
## text8  1960         1             1   1 (0.514)   0 (0.937)
## text14 1966         0             1   0 (0.897)   1 (0.746)
## text17 1969         1             0   0 (0.766)   0 (0.981)
## text18 1970         1             1   0 (0.816)   0 (0.852)
## text19 1971         1             0   0 (0.666)   0 (0.967)
## text23 1975         0             0   0 (0.887)   0 (0.982)
## text31 1983         1             1   0 (0.687)   0 (0.945)
## text34 1986         1             1   0 (0.739)   0 (0.967)
```

Where did we do well?

```r
all[11:21, ]
```

```
##        year coalition dist_category svm_results glm_results
## text35 1987         1             0   0 (0.847)   0 (0.668)
## text36 1988         1             0   0 (0.637)   1 (0.944)
## text37 1989         1             0   0 (0.884)   0 (0.968)
## text38 1990         1             1   0 (0.814)   0 (0.601)
## text40 1992         1             0   0 (0.697)   0 (0.982)
## text42 1994         1             0   0 (0.778)   0 (0.876)
## text44 1996         1             0   0 (0.805)   0 (0.983)
## text46 1998         1             0   0 (0.685)   0 (0.854)
## text49 2001         1             1    0 (0.63)   0 (0.993)
## text54 2006         1             0   0 (0.738)    1 (0.97)
## text59 2011         1             0   0 (0.715)   0 (0.959)
```

Cross-validation (SVM)

```r
cross_validate(pm_cont, 3, "SVM")
```

```
## Fold 1 Out of Sample Accuracy = 0.75
## Fold 2 Out of Sample Accuracy = 0.6190476
## Fold 3 Out of Sample Accuracy = 0.7

## [[1]]
## [1] 0.7500000 0.6190476 0.7000000
##
## $meanAccuracy
## [1] 0.6896825
```

Cross-validation (GLMNET)

```r
cross_validate(pm_cont, 3, "GLMNET")
```

```
## Fold 1 Out of Sample Accuracy = 0.9444444
## Fold 2 Out of Sample Accuracy = 0.7826087
## Fold 3 Out of Sample Accuracy = 1.25
```

```
## [[1]]
## [1] 0.9444444 0.7826087 1.2500000
##
## $meanAccuracy
## [1] 0.992351
```

Limitations

- How about the baseline?

```
prop.table(table(all$dist_category))
```

```
##
##         0         1
## 0.6666667 0.3333333
```

- How about substantive issues?
- And granularity?

Transparency!