Descriptive Analyses & Dictionaries

Computational Text Analysis

Theresa Gessler

University of Zurich | http://theresagessler.eu/ | @th_ges

2022-05-09

Program

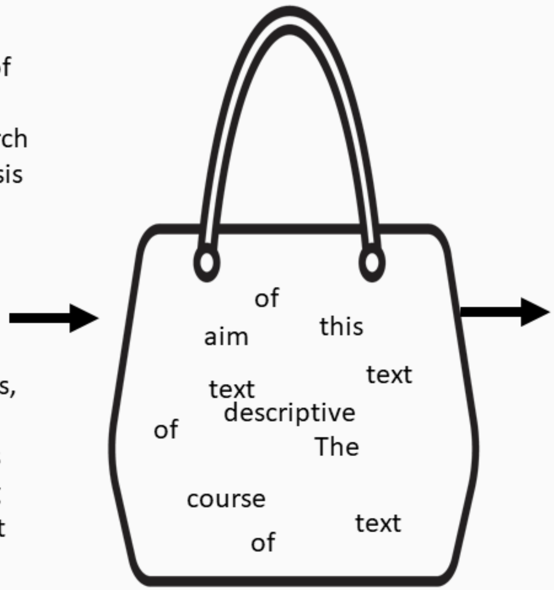
- **Workflow**
  - Text analysis Objects
- **Descriptive Analysis**
  - at corpus level: keywords in context, readability
  - at dfm level: keyness statistics
- **Dictionary analysis**
  - conceptually
  - in quanteda

Workflow

Workflow

The aim of this course is to introduce students to the quantitative analysis of textual data. We will cover both applications in recent empirical research and the implementation of text analysis techniques through hands-on experiences using the R statistical programming language.

The course will cover the collection of text data with webscraping techniques, text preprocessing, dictionaries and descriptive analysis of texts, as well as supervised and unsupervised learning methods to classify the content of text corpora.



the	7
of	6
text	4
to	3
analysis	3
.	3
and	3
,	3
course	2
data	2

Workflow

Three types of objects in quanteda:

- **corpus**
  - texts as strings with metadata in data frame
- **tokens**
  - separated individual features in list of vectors
  - more efficient but maintains the word order
- **document-feature matrix (dfm)**
  - Frequency of features per document in matrix / table format
  - most efficient structure, but no information about positions of the words ('bag of words')

Workflow

Example: US Presidential Debate

- 1st presidential debate bw/ Donald Trump & Joe Biden, moderated by Chris Wallace
- debate transcript with speakers and time stamps

2020 PRESIDENTIAL
DEBATE
CASE WESTERN RESERVE UNIVERSITY
AND CLEVELAND CLINIC

Transcript obtained from Kaggle: https://www.kaggle.com/headsortails/us-election-2020-presidential-debates

Corpus

Workflow

Corpus

In R (03_descriptive_analysis.rmd)

- **loading** 'us_election_2020_1st_presidential_debate.csv'
- **inspecting** the dataset: content, structure, variables
  - *bonus*: **wrangle**: generate a shorter speaker variable
- **creating the corpus**: use `corpus()` to create a `quanteda` corpus
  - *bonus*: specify useful names for each text in the corpus

```
first_debate <- read.csv("../data/us_election_2020_1st_presidential_debate.csv",
  stringsAsFactors = F,encoding="UTF-8")

# optional : speaker
first_debate <- first_debate %>%  mutate(speaker=str_extract(speaker,"[A-z]*$"))

debate_corp <- corpus(first_debate)

# optional : renaming
docnames(debate_corp) <- paste0(1:nrow(first_debate),"_",
                                first_debate$speaker)
```

Workflow

Corpus

- **corpus**: Structured collection of texts
  - Documents: Texts (by default: `text` variable - specify with `text_field=`)
  - Document variables / docvars: variables obtained from data set

```r
debate_corp[1:4]
```

```
## Corpus consisting of 4 documents and 2 docvars.
## 1_Wallace :
## "Good evening from the Health Education Campus of Case Wester..."
##
## 2_Wallace :
## "This debate is being conducted under health and safety proto..."
##
## 3_Biden :
## "How you doing, man?"
##
## 4_Trump :
## "How are you doing?"
```

Workflow

Summary of the corpus

```
summary(debate_corp) %>% head()
```

```
##        Text Types Tokens Sentences speaker minute
## 1 1_Wallace    88    135         8 Wallace  01:20
## 2 2_Wallace    83    116         5 Wallace  02:10
## 3   3_Biden     6      6         1   Biden  02:49
## 4   4_Trump     5      5         1   Trump  02:51
## 5   5_Biden     3      3         1   Biden  02:51
## 6 6_Wallace    89    149         9 Wallace  03:11
```

Important terms

- **Text**: each document of the corpus
- **Tokens**: total number of words in a text (or corpus), independent of repetitions
- **Types**: Number of different words in a text (or corpus)

Tokens

Workflow

Tokens

- **individual features**, stored in list of vectors
- more efficient format than corpus but retains the word order
  - *'chop' the sentences without 'shaking' the bag

Use

- **some of the analysis on corpus** (e.g. Keywords in Context)
- **pre-processing** (also at dfm-level)
  - removing irrelevant features, manipulation of features
  - *advantage of tokens*: word order provides context
- **Dictionaries** (also at dfm-level)
  - *advantage of tokens*: multi-word expressions, word order as context

→ **What constitutes a feature (word, n-gram, sentence, letter)?**

→ **Which of these features are relevant? How do I prepare them?**

Workflow

Tokenization

- separation into features is called **tokenization** (command: `tokens()`)
- possible at different levels: word, sentence or character.

```
tokens(debate_corp, what="word")[[1]][1:10]
```

```
##  [1] "Good"      "evening"   "from"      "the"       "Health"    "Education"
##  [7] "Campus"    "of"        "Case"      "Western"
```

```
tokens(debate_corp, what="character")[[1]][1:10]
```

```
##  [1] "G" "o" "o" "d" "e" "v" "e" "n" "i" "n"
```

Default: word-level tokenization

```
debate_toks <- tokens(debate_corp)
```

→ We return to tokens later for pre-processing and dictionaries

Document feature matrix

Document feature matrix

- **frequency of features per document in matrix format**
  - created from corpus or tokens
- most efficient structure, but no information on word positions → **'bag of words'**
- origin for most statistical analyses
  - combination of word frequency with document variables

```
debate_dfm <- dfm(debate_toks)
debate_dfm
```

```
## Document-feature matrix of: 789 documents, 2,297 features (99.16% sparse) and 2
docvars.
##              features
## docs          good evening from the health education campus of case western
##    1_Wallace    1       1    2  15      1         1      1  5    1       1
##    2_Wallace    0       0    0  10      2         0      0  1    0       0
##    3_Biden      0       0    0   0      0         0      0  0    0       0
##    4_Trump      0       0    0   0      0         0      0  0    0       0
##    5_Biden      0       0    0   0      0         0      0  0    0       0
##    6_Wallace    0       0    0  10      0         0      0  3    0       0
```

Descriptive Analysis

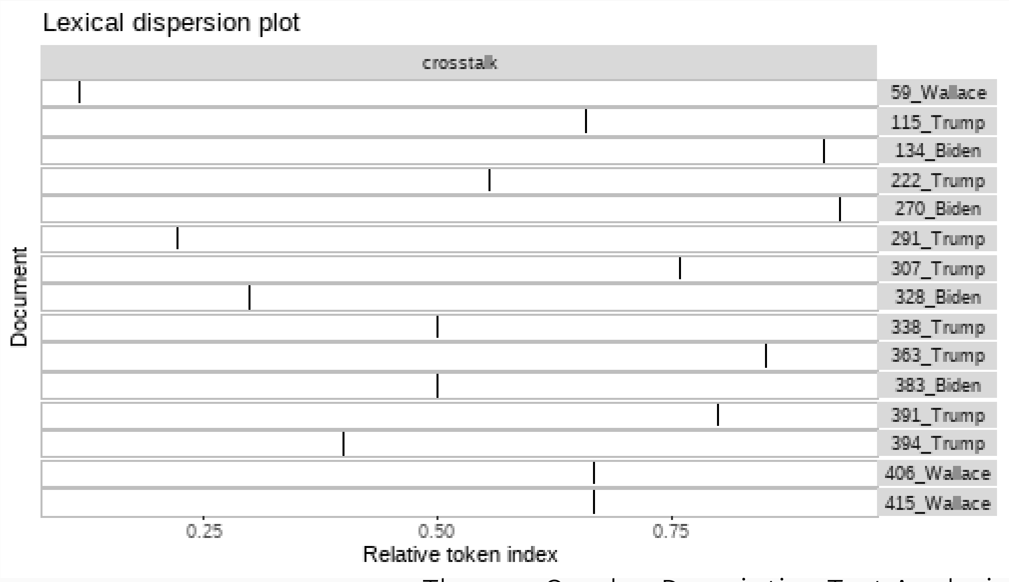
Descriptive Analysis

you can follow along in R: 03_descriptive_analysis.rmd

Where are terms used?

e.g. when do interruptions happen?

```
kwic(debate_corp, "crosstalk") %>% head(15) %>%
  textplot_xray()
```



In which context are terms used?

Keywords in context, e.g. 'country'

```
kwic(debate_corp, "country",window=4) %>%
  head()
```

```
## Keyword-in-context with 6 matches.
##    [167_Trump, 9]            to you, the | country | would have been left
##  [167_Trump, 150] should have closed our | country | . Wait a minute
##    [169_Trump, 9] should have closed our | country | because you thought it
##   [215_Trump, 36]     the history of our | country | . And by the
##    [226_Trump, 9]      to shut down this | country | and I want to
##   [228_Trump, 29]       to shut down the | country | . We just went
```

at tokens-level: after removing stopwords

Descriptive Analysis

How are the texts written?

- e.g. readability statistics at text level

```
textstat_readability(debate_corp) %>% head(3)
```

```
##     document   Flesch
## 1 1_Wallace 62.15573
## 2 2_Wallace 50.10547
## 3   3_Biden 97.02500
```

Paper: Schoonvelde et.al. (2019) "Liberals Lecture, Conservatives Communicate: Analyzing Complexity and Ideology in 381,609 Political Speeches." PLOS ONE 14, no. 2

Paper: Spirling (2015). "Democratization and Linguistic Complexity: The Effect of Franchise Extension on Parliamentary Discourse, 1832–1915." The Journal of Politics 78 (1): 120–36.

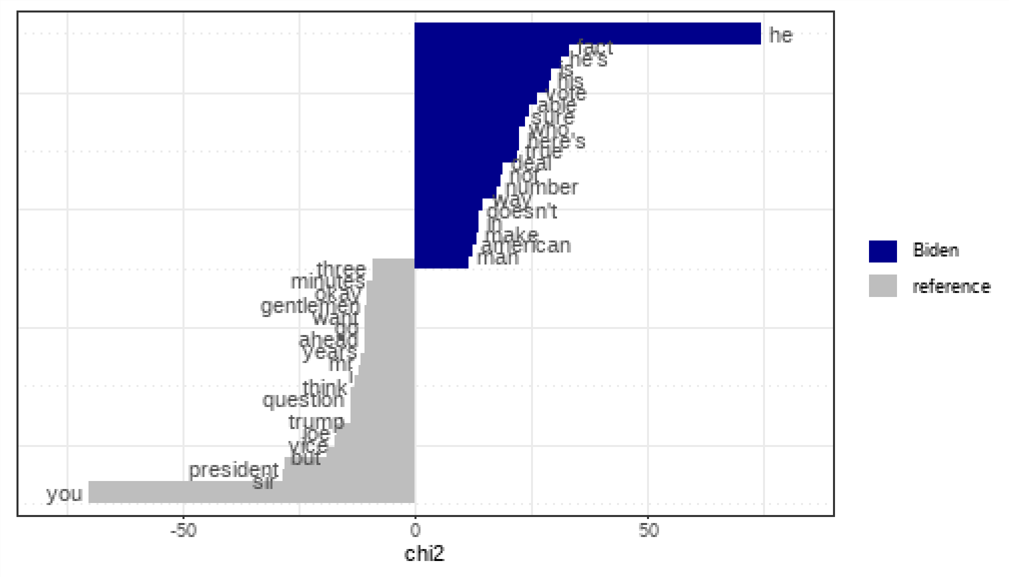
- e.g. frequent word combinations: `textstat_collocations()`

Descriptive Analysis

Which words are characteristic for each speaker?

- **at the dfm-level**: centered on *frequency of features*
- keyness of each term for speaker: `textstat_keyness()` with chi^2 or other measures

```
dfm_group(debate_dfm,speaker) %>% textstat_keyness("Biden") %>% textplot_keyness()
```



- other dfm-level statistics: `textstat_lexdiv()` (lexical diversity)

For next session: Practice & analysis

Complete 03_descriptive_analysis.rmd

- readability comparison
- Keywords in context
- keyness statistics

Dictionaries

Dictionaries

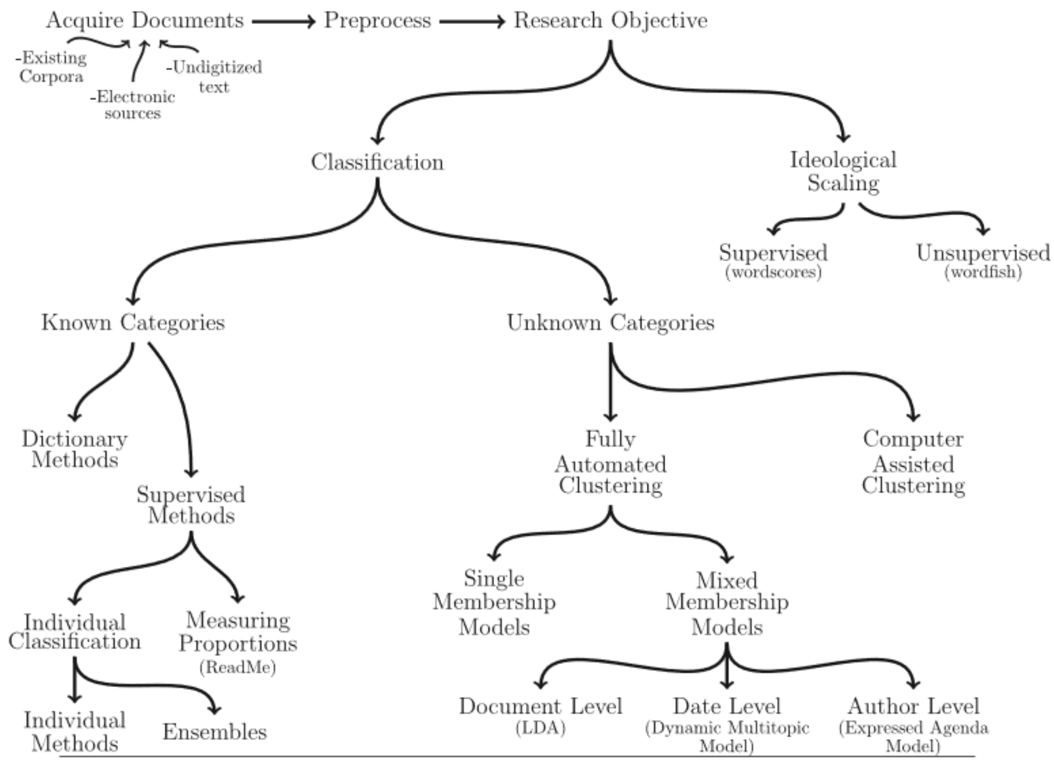


Fig. 1 An overview of text as data methods.

Grimmer, J. and B. M. Stewart (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis 21, 267-297.

Dictionaries

Purpose

- sorting text into categories
  - e.g.: immigration-related texts
- measuring degrees of certain characteristics
  - e.g. sentiment of amazon reviews
- finding the texts we care about
  - e.g. finding news articles about protests so that we can read them

Dictionaries

Degree of human involvement

- Human coding (100% human involvement)
    - *maybe something you did as a student?*
- **Supervised** (1-99% human involvement)
    - sorting data into known categories
- Unsupervised (0% human involvement)
    - automated sorting of data into unknown categories

We dicuss two methods of supervised classification

- with a dictionary
- with machine learning (tomorrow)

Dictionaries

A dictionary

A list of...

- 'keys', that stand for specific meanings or concepts
  - derived from theoretical considerations
- 'values' as empirical indicators of these keys

e.g. **family members** (*key*): mother, father, brother, sister, aunt, uncle, boyfriend, girlfriend, ... (*values*)

Measurement

- measurement of concept by frequency count of dictionary features
- more complex counts possible
  - and / or matches
  - continuous or binary measures of mentions

Dictionaries

Advantages

- easy to apply
- easy to adjust
- cost-efficient
- perfectly reliable (compared to human coding)

Disadvantages

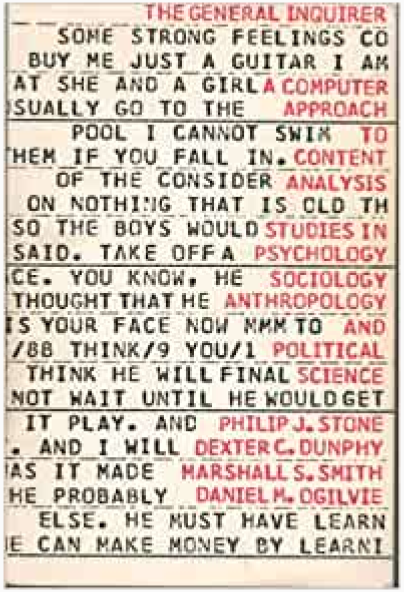
- rather supervised technique (human involvement)
- dependence on single words
  - esp. for small data: big effects
  - negations, dependency structures etc.
- applying dictionaries is difficult
  - context dependency
  - evolution of language
- creating dictionaries is difficult
  - theoretical considerations
  - exhaustiveness (see King, Lam and Roberts 2017)

→ A good dictionary is **exhaustive** but its values are also **unambiguous** (and possibly time-insensitive, context-relevant, …)

Existing dictionaries

Due to the long tradition of dictionary-research, many exist ready for use - for example...

- General Inquirer:
  - 182 categories
  - e.g. "self-references," "negatives"
- NRC Emotion Lexicon (english)
  - eight basic emotions
- Linguistic Inquiry and Word Count:
  - 82 language dimensions,
  - 4,500 words and stems
- newsmap
  - geographic locations
- and many others



First edition of the General Inquirer, 1966

Dictionaries

Ideologies - Pauwels (2011)

Measuring Populism: A Quantitative Text Analysis of Party Literature in Belgium. *Journal of Elections, Public Opinion and Parties* 21(1): 97-119.

Table A2. Dictionary

Dictionary	Dutch words	Translation
Conservatism	christ*; geloof; gezin; kerk; normen; porn*; seks*; waarden	christ*; belief; family; church; norm; porn*; sex*; values
Environment	ecol*; groene*; klimaat*; milieu*; opwarming	ecol*; green*; climate*; environment*; heating
Immigration	marok*; turk; allocht*; asiel*; halal*; hoofddoek*; illega*; immigr*; islam*; koran; moslim*; vreemd*	moroc*; turk; allocht*; asylum*; halal*; scarf*; illega*; immigr*; islam*; koran; muslim*; foreign*

→ uses frequency of word use to measure if text expresses ideology

Dictionaries

Recommendation Language - Schmader et al. (2007)

A Linguistic Comparison of Letters of Recommendation for Male and Female Chemistry and Biochemistry Job Applicants. *Sex roles* 57(7-8): 509–514.

Study-Defined Dimension Dictionaries Standout words: excellen*, superb, outstanding, unique, exceptional, unparalleled, *est, most, wonderful, terrific*, fabulous, magnificent, remarkable, estraordinar*, amazing, supreme*, unmatched

Ability words: talent*, intell*, smart*, skill*, ability, genius, brilliant*, bright*, brain*, aptitude, gift*, capacity, propensity, innate, flair, knack, clever*, expert*, proficient*, capable, adept*, able, competent, natural*, inherent*, instinct*, adroit*, creative*, insight*, analytical

Grindstone words: hardworking, conscientious, depend*, meticulous, thorough, diligen*, dedicate, careful, reliab*, effort*, assiduous, trust*, responsib*, methodical, industrious, busy, work*, persist*, organiz*, disciplined

Teaching words: teach, instruct, educat*, train*, mentor, supervis*, adviser, counselor, syllabus, syllabus, course*, class, service, colleague, citizen, communicate*, lectur*, student*, present*, rapport

Research words: research*, data, study, studies, experiment*, scholarship, test*, result*, finding*, publication*, publish*, vita*, method*, scien*, grant*, fund*, manuscript*, project*, journal*, theor*, discover*, contribution*

Note. * indicates that any word containing the letter string that precedes or follows the asterisk should be counted.

Applying and creating dictionaries

Integredient 1: Text

→ Examples on the Presidential Debate Corpus

- geographic: Which regions of the world are mentioned in the debate?
  - description
- thematic: how well can we predict the topic of a statement?
  - prediction

Follow along in R using 03_dictionaries.rmd

Applying and creating dictionaries

Ingredient 2: Dictionary

```
newsmap_dict <- dictionary(file = "english.yml",
                           format = "YAML")
```

keys (e.g. Africa) are translated into **values** (e.g. addis ababa)

```
print(newsmap_dict)
```

```
## Dictionary object with 5 primary key entries and 3 nested levels.
## - [AFRICA]:
##   - [EAST]:
##     - [BI]:
##       - burundi, burundian*, bujumbura
##     - [DJ]:
##       - djibouti, djiboutian*
##     - [ER]:
##       - eritrea, eritrean*, asmara
##     - [ET]:
##       - ethiopia, ethiopian*, addis ababa
##     - [KE]:
##       - kenya, kenyan*, nairobi
```

Applying the dictionary - dfm

```
dfm_lookup(debate_dfm,newsmap_dict)[650:655,111:113]
```

```
## Document-feature matrix of: 6 documents, 3 features (94.44% sparse) and 2 docvars.
##              features
## docs           AMERICA.NORTH.GL AMERICA.NORTH.PM AMERICA.NORTH.US
##   650_Biden                   0                0                0
##   651_Trump                   0                0                0
##   652_Wallace                 0                0                0
##   653_Trump                   0                0                0
##   654_Wallace                 0                0                0
##   655_Biden                   0                0                1
```

→ lookup command **looks up** dictionary values and converts them to keys

→ results match our concepts, not the values

Applying and creating dictionaries

Applying the dictionary - Tokens

```
tokens_lookup(debate_toks,newsmap_dict)[650:655]
```

```
## Tokens consisting of 6 documents and 2 docvars.
## 650_Biden :
## character(0)
##
## 651_Trump :
## character(0)
##
## 652_Wallace :
## character(0)
##
## 653_Trump :
## character(0)
##
## 654_Wallace :
## character(0)
##
## 655_Biden :
## [1] "AMERICA.NORTH.US"
```

Getting aggregate statistics

We can obtain frequencies with `textstat_frequency()`

```
dfm_lookup(debate_dfm,newsmap_dict) %>% textstat_frequency()
```

```
##              feature frequency rank docfreq group
## 1   AMERICA.NORTH.US        44    1      35   all
## 2       ASIA.EAST.CN        10    2       9   all
## 3     EUROPE.EAST.RU         6    3       6   all
## 4     EUROPE.WEST.FR         5    4       4   all
## 5      ASIA.SOUTH.IN         2    5       2   all
## 6  AMERICA.CENTER.MX         1    6       1   all
## 7   AMERICA.SOUTH.BR         1    6       1   all
## 8       ASIA.EAST.JP         1    6       1   all
## 9       ASIA.WEST.IQ         1    6       1   all
## 10    EUROPE.EAST.UA         1    6       1   all
## 11   EUROPE.NORTH.IE         1    6       1   all
## 12    EUROPE.WEST.DE         1    6       1   all
```

→ **How often are countries mentioned?**

Applying and creating dictionaries

In R

03_dictionaries.rmd, line 69 ff.

- load the newsmap dictionary
- apply the newsmap dictionary to the dfm
- apply the newsmap dictionary to the tokens and then create a dfm
- compare the output of `textstat_frequency()` for both objects: **Why is there a difference?**

Dictionaries for dfms and tokens

```
newsmap_dict <- dictionary(file = "english.yml",
                           format = "YAML")
debate_dfm %>% dfm_lookup(newsmap_dict) %>% textstat_frequency() %>% head(4)
```

```
##           feature frequency rank docfreq group
## 1 AMERICA.NORTH.US        44    1      35   all
## 2     ASIA.EAST.CN        10    2       9   all
## 3   EUROPE.EAST.RU         6    3       6   all
## 4   EUROPE.WEST.FR         5    4       4   all
```

```
debate_toks %>% tokens_lookup(newsmap_dict) %>% dfm() %>% textstat_frequency() %>%
head(4)
```

```
##           feature frequency rank docfreq group
## 1 america.north.us        58    1      40   all
## 2     asia.east.cn        10    2       9   all
## 3   europe.east.ru         6    3       6   all
## 4   europe.west.fr         5    4       4   all
```

Applying and creating dictionaries

→ **Some of the dictionary keys contain multi-word expressions** which depend on *word order* - e.g. the entry for America

```
newsmap_dict$AMERICA$NORTH$US
```

```
## [1] "united states" "us"            "american*"     "washington"
## [5] "new york"
```

Multi-word entries remain intact in the tokens but are cut apart in the dfm

```
tokens_select(debate_toks,newsmap_dict)[12]
```

```
## Tokens consisting of 1 document and 2 docvars.
## 12_Biden :
## [1] "American" "United"   "States"   "United"   "States"   "American"
```

```
debate_toks[12] %>% dfm() %>% dfm_select(newsmap_dict)
```

```
## Document-feature matrix of: 1 document, 1 feature (0.00% sparse) and 2 docvars.
##           features
## docs        american
##   12_Biden        2
```

Working with dictionary results

Potential questions

- How often are specific concepts mentioned?
- Are specific concepts mentioned?
- How do these mentions develop, dependent on y (e.g. time, speaker, ...)

→ **We need to work with the results!**

→ One way to do so is to **weigh the results**

```
geography_dfm <- debate_toks %>%
                   tokens_lookup(newsmap_dict) %>%
                   dfm()
```

Working with dictionary results

Weighting

- the frequency of a concept → continuous per text

```
geography_dfm %>% textstat_frequency() %>% head(2)
```

```
##            feature frequency rank docfreq group
## 1 america.north.us        58    1      40   all
## 2     asia.east.cn        10    2       9   all
```

- the presence of a concept (0 / 1 per text)

```
geography_dfm %>% dfm_weight("boolean") %>%
 textstat_frequency() %>% head(2)
```

```
##            feature frequency rank docfreq group
## 1 america.north.us        40    1      40   all
## 2     asia.east.cn         9    2       9   all
```

Working with dictionary results

Weighting

- a proportion
- use `prop``weighting before the`lookup`command or specify a`nomatch` argument so the dictionary so the proportions relate to all words, not the dictionary features

```
tokens_lookup(debate_toks, newsmap_dict,nomatch =
"NN") %>%
  dfm() %>% dfm_group(speaker) %>%
  dfm_weight("prop") %>%
textstat_frequency(group=speaker) %>% head()
```

```
##             feature    frequency rank docfreq group
## 1                nn 0.9945750452    1       1 Biden
## 2  america.north.us 0.0042624645    2       1 Biden
## 3    europe.west.fr 0.0003874968    3       1 Biden
## 4      asia.east.cn 0.0002583312    4       1 Biden
## 5 america.center.mx 0.0001291656    5       1 Biden
## 6  america.south.br 0.0001291656    5       1 Biden
```

Working with dictionary results

Interpreting dictionaries

When you're done with reshaping the results, most people find it easier to work with data frames

→ you can use `convert("data.frame")` to convert the dfm into a data frame → Use in statistical analysis

```
dfm_lookup(debate_dfm,newsmap_dict) %>%
   convert("data.frame") %>%
   head()
```

```
##      doc_id AFRICA.EAST.BI AFRICA.EAST.DJ AFRICA.EAST.ER AFRICA.EAST.ET
## 1 1_Wallace              0              0              0              0
## 2 2_Wallace              0              0              0              0
## 3   3_Biden              0              0              0              0
## 4   4_Trump              0              0              0              0
## 5   5_Biden              0              0              0              0
## 6 6_Wallace              0              0              0              0
##   AFRICA.EAST.KE AFRICA.EAST.KM AFRICA.EAST.MG AFRICA.EAST.MU AFRICA.EAST.MW
## 1              0              0              0              0              0
## 2              0              0              0              0              0
```

Dictionaries: creation & evaluation

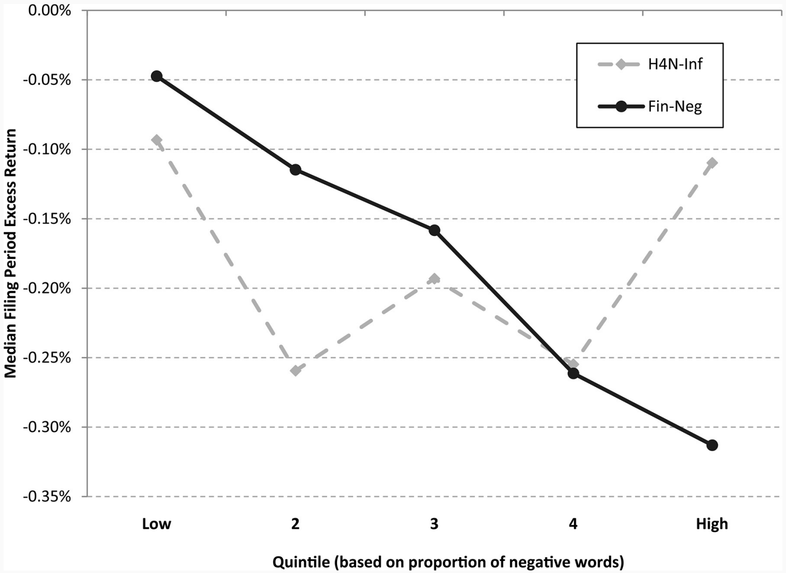
Dictionaries: creation & evaluation

Which words signal that concept is being used?

Loughran, T. and McDonald, B. (2011), When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. The Journal of Finance, 66: 35-65. doi:10.1111/j.1540-6261.2010.01625.x

- "In a large sample of 10-Ks during 1994 to 2008, almost three-fourths of the words identified as negative by the widely used Harvard Dictionary are words typically not considered negative in financial contexts."
- examples
    - costs, tax, expense, board, foreign, vice, decrease, risks, ...

Dictionaries: creation & evaluation



Source: Loughran, T. and McDonald, B. (2011), When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. The Journal of Finance, 66: 35-65. doi:10.1111/j.1540-6261.2010.01625.x

Dictionaries: creation & evaluation

Creating a dictionary

For creating your own dictionary:

- remember creating dictionaries is difficult & humans are bad at it
- try to come up with as many possible ways to address your concept as possible
    - use your imagination, ask others, use synonym lexicons...
- test whether the words are really used in connection to the concept

Dictionaries: creation & evaluation

Creating a dictionary with quanteda

- if you need to create a dictionary from scratch or edit an existing dictionary, you can define dictionaries as **lists of words**

```
simple_dict <- dictionary(list(liberalism = c('*tax*', '*reduction*', 'bureaucrat*',
'compet*', 'dereg*',
'effici*',  'job*', 'tax')))

print(simple_dict)
```

```
## Dictionary object with 1 key entry.
## - [liberalism]:
##   - *tax*, *reduction*, bureaucrat*, compet*, dereg*, effici*, job*, tax
```

more options, such as reading in files, are described in the quanteda documentation - *if you actually want to do this for your thesis, I recommend working with an excel file or similar*

Form: glob patterns and regular expressions

- often you want dictionaries to be more universal - for example, to capture words regardless of endings or with different spellings
  - e.g. student, students

- **glob patterns**: wildcard characters, see wikipedia)
  - example: Pauwels (2011): **christ*** → captures: *Christian, Christ, Christianty etc.*
  - * matches any string of characters
  - ? matches exactly one character
  - [ ] matches one character given in the bracket, e.g [AB] -> matches **A** or **B** → e.g. "r[au]n" for run and ran
- more complex, but also more powerful: **regular expressions / regex**
  - regex cheat sheet, another regex cheat sheet

Dictionaries: creation & evaluation

Evaluation

- evaluation of dictionaries is crucial to **validate** the measures
    - in which context are words used?
    - do I find all the texts that are relevant?

→ formal procedures for supervised learning

→ more informal procedures to get an impression of the text

Dictionaries: creation & evaluation

Evaluation

- **Use "extreme" texts**:
  - e.g. how left and right politicians speaking about an issue
  - 5-stars and 1-star ratings of a product
  - policy uncertainty in times of crisis and in times of boom

→ see if the measure behaves as you would expect it to

Evaluation

- **Identify frequent matches and explore their context**
  - use `tokens_select()` to find frequent matches
  - explore context e.g. with the `kwic()`-function()

```
debate_toks %>%
  tokens_select(newsmap_dict) %>%
  dfm() %>%
  topfeatures(8)
```

```
##  american        us    united    states     china americans     paris       new
##        19        19        10        10        10         6         5         4
```

→ `dfm_select()` and `tokens_select()` do not convert values into dictionary keys, they just discard everything else

Homework: Applying and creating dictionaries

Complete 03_dictionaries.rmd

- evaluating dictionary results by group
- applying specific levels of a dictionary
- use weighting with dictionaries
- create your own dictionary to measure a different topic
- use the dictionary to classify texts into topics by finding a decision rule
- transfer to EUI theses

Literature

- Muddiman, Ashley, Shannon C. McGregor, and Natalie Jomini Stroud. "(Re)Claiming Our Expertise: Parsing Large Text Corpora With Manually Validated and Organic Dictionaries." Political Communication 0, no. 0 (November 7, 2018): 1–13.
- Loughran, Tim, and Bill Mcdonald. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks." The Journal of Finance 66, no. 1 (2011): 35–65.

Tomorrow

Tomorrow

What we'll cover

- **Supervised Classification**
  - using labelled data to learn about new data
  - from pre-processed data to results
- **evaluation techniques**
  - also relevant for dictionaries
- classification accuracy as substantive information
  - using predicted labels to infer quantities of interest
  - example: measuring polarization / measuring gender differences
- maybe: other statistical methods
  - wordscores, wordfish
- **packages**: `quanteda`, `quanteda.textmodels`, `caret`

Tomorrow

What we'll cover

- **Unsupervised Classification**
  - topic models
  - cluster analysis
  - using the structural topic model
- elements of weak supervision
  - supervised topic models
  - latent semantic scaling
- maybe: other statistical methods
  - wordscores, wordfish
- **packages**: `stm`, (...)

Tomorrow

Preparation

- **complete:**
  - 01_rmarkdown.rmd
  - 01_textanalysis.rmd
  - 02_transform_preproc.rmd → pre-processing techniques
  - 02_descriptive_analysis.rmd
  - 02_dictionaries.rmd
- if you want, do the additional exercises with your own data

Tomorrow

Preparation

Building on the course

- think of your data and your concept
    - is there any labelled data you could use?
        - e.g. pre-coded data
    - what would you want to find in unlabelled data?
- could you use classification to study differences between (binary) groups
    - e.g. parties, partisans, genders, ...
- is there a text corpus that you found interesting but you have very limited knowledge of?
    - e.g. a data archive
- is there a corpus of highly similar texts where you are interested in framing?
    - e.g. open survey questions

Tomorrow

Literature

Pre-processing

- Denny, Matthew J., und Arthur Spirling. „Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It“. Political Analysis 26, Nr. 2 (April 2018): 168–89. https://doi.org/10.1017/pan.2017.44.

Dictionaries

- Muddiman, Ashley, Shannon C. McGregor, and Natalie Jomini Stroud. “(Re)Claiming Our Expertise: Parsing Large Text Corpora With Manually Validated and Organic Dictionaries.” Political Communication 0, no. 0 (November 7, 2018): 1–13. https://doi.org/10.1080/10584609.2018.1517843.
- Loughran, Tim, and Bill Mcdonald. “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks.” The Journal of Finance 66, no. 1 (2011): 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x.

Literature

Classification

- Barberá, Pablo, Amber E. Boydstun, Suzanna Linn, Ryan McMahon, and Jonathan Nagler. "Automated Text Classification of News Articles: A Practical Guide." Political Analysis, undefined/ed, 1–24. https://doi.org/10.1017/pan.2020.8.
- Peterson, Andrew, and Arthur Spirling. "Classification Accuracy as a Substantive Quantity of Interest: Measuring Polarization in Westminster Systems." Political Analysis 26, no. 1 (January 2018): 120–28. https://doi.org/10.1017/pan.2017.39.
- Beltran, Javier, Aina Gallego, Alba Huidobro, Enrique Romero, and Lluís Padró. "Male and Female Politicians on Twitter: A Machine Learning Approach." European Journal of Political Research n/a, no. n/a. Accessed March 24, 2020. https://doi.org/10.1111/1475-6765.12392.
- Cranmer, Skyler J. "Introduction to the Virtual Issue: Machine Learning in Political Science," n.d., 9.

Literature

Topic models

- DiMaggio, Paul, Manish Nag, and David Blei. "Exploiting Affinities between Topic Modeling and the Sociological Perspective on Culture: Application to Newspaper Coverage of U.S. Government Arts Funding." Poetics, Topic Models and the Cultural Sciences, 41, no. 6 (December 2013): 570–606. https://doi.org/10.1016/j.poetic.2013.08.004.
- Roberts, Margaret E., Brandon M. Stewart, and Dustin Tingley. "Stm: R Package for Structural Topic Models." Journal of Statistical Software, 2013.
- Bauer, Paul C., Pablo Barberá, Kathrin Ackermann, and Aaron Venetz. "Is the Left-Right Scale a Valid Measure of Ideology?" Political Behavior 39, no. 3 (2017): 553–83.
- Egami, Naoki, Christian J Fong, Justin Grimmer, Margaret E Roberts, and Brandon M Stewart. "How to Make Causal Inferences Using Texts∗," n.d., 68.
- Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. "Structural Topic Models for Open-Ended Survey Responses." American Journal of Political Science 58, no. 4 (October 1, 2014): 1064–82. https://doi.org/10.1111/ajps.12103.

Thank you! - Questions?