PLSC 504: Some Mostly-Random Text Analysis Methods

December 9, 2020

Le Menu

- Dictionary-based methods (including sentiment analysis)
- Topic models
- Text scaling

Overview: Dictionary-Based Methods

- **Classification** task:
  - *Categorize* documents into classes C, and/or
  - *Score* documents degree of association with those classes.

- Heuristic: **Dictionaries assign weights to words / terms.**

- Formally: For $j \in \{1...J\}$ words in a corpus of $i = \{1...N\}$ documents, the *document score* is:

$$S_i = \frac{\sum_{j=1}^{J} \omega_j X_{ij}}{\sum_{j=1}^{J} X_{ij}}$$

  where
  - X_{ij} is the number of instances of word j in document i, and
  - ω_j is the weight assigned to each word by the dictionary.

General Dictionary-Based Methods: How-To

1. Obtain / preprocess documents (stemming, stop words, etc.)
2. Obtain / create a dictionary
3. Score documents by calculating S_i
   - Weights ω_j can be positive or negative
   - Words in the corpus but not in the dictionary have $\omega_j = 0$
4. (Optional:) Classify documents by mapping $S_i \rightsquigarrow C_i$

Toy Example: "Truthiness"

- Document: {TRUE FALSE TRUE FALSE TRUE}
- Dictionary:

Term	ω_j
TRUE	1.0
FALSE	0.0

- Word counts:

$$\mathbf{X} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

- Score:

$$S_i = \frac{(1.0 \times 3) + (0.0 \times 2)}{(3 + 2)} = \frac{3}{5} = 0.6$$

Dictionary-Based Classification Tasks

- Topic(s)
  - What are documents *about*?
  - What thing(s) are *emphasized*?
- Sentiment
  - What is the *emotional valence* of the documents?
  - What are the emotions expressed? (pity, anger, jealousy, etc.)
- Tone / Style
  - Authorship / provenance
  - Specialization of language (e.g., "hold harmless")

Sentiment Analysis

"...[C]omputational study of how opinions, attitudes, emotions, and perspectives are expressed in language..."

– Liu (2011)

Lots of research in computer science and linguistics: Pang and Lee (2004, 2008), Tong (2001), Zhou, Chen and Wang (2010), Das and Chen (2001), Dasgupta and Ng (2009), Pang et al. (2002), Turney (2002), Wiebe (2000), Shanahan, Qu, and Wiebe (2006), Jindal and Liu (2006), Liu (2006), Nigam and Hurst (2005), Hu and Liu (2004), Choi and Cardie (2010), and many, many more...

A good overview is:

Pang, Bo, and Lillian Lee. 2008. "Opinion Mining and Sentiment Analysis." Foundations and Trends in Information Retrieval 2:1-135.

Where Do (Sentiment) Dictionaries Come From?

- "Standard" dictionaries
  - Code sentiment in common (contemporary, usually American) English
  - See below; there's a list here
- "By hand"...
  - Requires careful thought / luck / divine help
  - **Validate**. Seriously.
- "Crowdsourced" methods: RAs, MTurk, etc.
  - "On a scale from -10 to 10, how positive is the word...?"
  - Can be made context-specific, etc.
- Statistical approaches
  - Fit a model to some document-level outcome $\rightarrow$ most predictive words = dictionary
  - "Model" = lasso / ridge regression / elastic net, etc.
  - Again, **validation** is key...

Common (English) Sentiment Dictionaries

- General Inquirer (`http://www.wjh.harvard.edu/~inquirer/`)
- AFINN (`http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010`)
- QDAP dictionaries (`https://cran.r-project.org/web/packages/qdap/index.html`)
- WordStat (find it here)
- LIWC (`http://liwc.wpengine.com/`)

Sentiment Dictionary Examples

General Inquirer:

- Words scored either positive ($+1$) or negative (-1)
- 1637 positive words, 2005 negative words

AFINN (2477 total words, scored [-5,5]):

Term	ω_j
`bastard`	-5
`bitch`	-5
⋮	⋮
`worn`	-1
`some kind`	0
`aboard`	1
⋮	⋮
`superb`	5
`thrilled`	5

Sentiment Analysis Options in R

- `SentimentAnalysis`
  - Built by finance people → dictionaries, etc.
  - Plays well with `tm`
  - My current favorite (see the vignette)

- `tidyverse`, etc.
  - Requires admission to the cult of Wickham
  - Details here: `https://www.tidytextmining.com/`
  - Tons of tutorials (here, here, here, etc.)

- `RSentiment` (super minimal)

- `sentiment` (deprecated)

`SentimentAnalysis` Details

- Works with character objects, data frames, corpuses / TDMs / DTMs from `tm`
- Built-in dictionaries: General Inquirer, QDAP, two finance-specific (Henry 2008; Loughran and McDonald 2011)
- Can also create dictionaries "by hand" or through predictive power of words vis-a-vis some response (via glm, lasso, etc.)
- `analyzeSentiment` is the workhorse
  - Defaults to using all four built-in dictionaries
  - Stems and removes stop words by default
  - Outputs a `data.frame` with document-level sentiment scores
- Other useful things:
  - Built-in tokenizer / N-gram creator
  - Convert continuous sentiment scores to binary (0/1) or directional (-1/0/1) values
  - Can generate predictions and assess predictive performance...

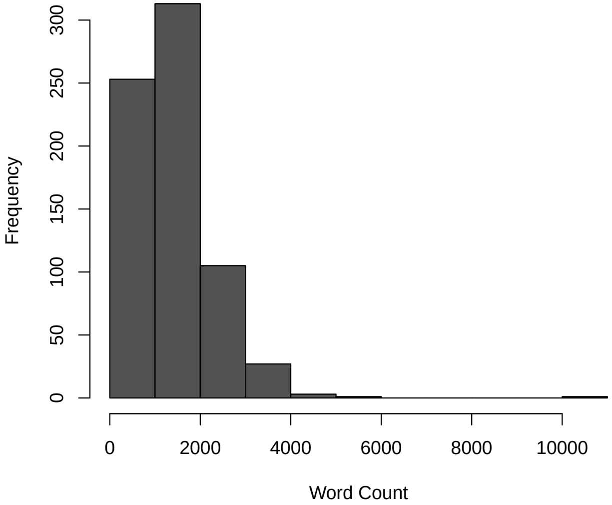
Example: UNHCR Speeches

- All speeches made by the High Commissioner of the U.N. Refugee Agency, 1970-2016 ($N = 703$)
- Metadata include ID, speaker, title, and date
- Source: `https://www.kaggle.com/franciscadias/un-refugee-speech-analysis/`

UNHCR Speeches...

```
> UN <- read.csv(text=temp,
+                 stringsAsFactors=FALSE,allowEscapes=TRUE)
> rm(temp)
>
> UN$content <- removeNumbers(UN$content) # no numbers
> UN$content <- str_replace_all(UN$content, "[\n]", " ") # line breaks
> UN$content <- removeWords(UN$content,stopwords("en")) # remove stopwords
> UN$Year <- as.numeric(str_sub(UN$by, -4)) # Year of the speech
> UN$foo <- str_extract(UN$by, '\\b[^,]+$')
> UN$Date <- as.Date(UN$foo, format="%d %B %Y") # date of speech
> UN$foo <- NULL
> UN$Author <- "Goedhart"  # Fix names...
.
.
.
> # Corpus:
>
> UN2 <- with(UN, data.frame(doc_id = id,
+                             text = content))
> ds <- DataframeSource(UN2)
> UNC <- Corpus(ds)
> meta(UNC)
data frame with 0 columns and 703 rows
>
> # Some tools in SentimentAnalysis...
>
> UNCount<-countWords(UNC,removeStopwords=FALSE)
> summary(UNCount$WordCount)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     50     762    1283    1404    1864   10948
```

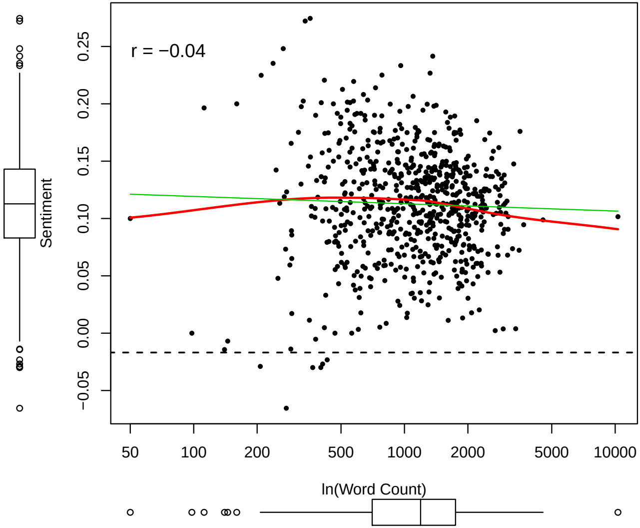
UNHCR Speech Word Counts, 1970-2016



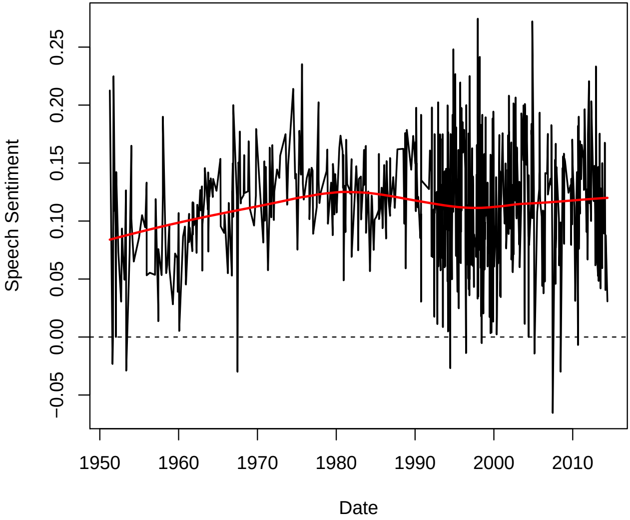
Simple Sentiment Analysis

```
> UNSent <- analyzeSentiment(UNC)

> summary(UNSent)
   WordCount       SentimentGI        NegativityGI     PositivityGI    SentimentHE
 Min.   :   50   Min.   :-0.065   Min.   :0.002   Min.   :0.00   Min.   :-0.011
 1st Qu.:  703   1st Qu.: 0.083   1st Qu.:0.115   1st Qu.:0.23   1st Qu.: 0.011
 Median : 1193   Median : 0.113   Median :0.134   Median :0.25   Median : 0.017
 Mean   : 1299   Mean   : 0.113   Mean   :0.135   Mean   :0.25   Mean   : 0.017
 3rd Qu.: 1747   3rd Qu.: 0.143   3rd Qu.:0.154   3rd Qu.:0.27   3rd Qu.: 0.022
 Max.   :10306   Max.   : 0.275   Max.   :0.237   Max.   :0.36   Max.   : 0.072
  NegativityHE      PositivityHE      SentimentLM       NegativityLM     PositivityLM
 Min.   :0.0000   Min.   :0.000   Min.   :-0.119   Min.   :0.000   Min.   :0.000
 1st Qu.:0.0043   1st Qu.:0.019   1st Qu.:-0.043   1st Qu.:0.045   1st Qu.:0.026
 Median :0.0070   Median :0.024   Median :-0.024   Median :0.057   Median :0.032
 Mean   :0.0075   Mean   :0.025   Mean   :-0.027   Mean   :0.060   Mean   :0.032
 3rd Qu.:0.0101   3rd Qu.:0.029   3rd Qu.:-0.009   3rd Qu.:0.073   3rd Qu.:0.038
 Max.   :0.0249   Max.   :0.072   Max.   : 0.044   Max.   :0.136   Max.   :0.068
 RatioUncertaintyLM SentimentQDAP    NegativityQDAP  PositivityQDAP
 Min.   :0.000      Min.   :-0.066   Min.   :0.000   Min.   :0.003
 1st Qu.:0.011      1st Qu.: 0.064   1st Qu.:0.056   1st Qu.:0.144
 Median :0.014      Median : 0.084   Median :0.075   Median :0.160
 Mean   :0.015      Mean   : 0.084   Mean   :0.076   Mean   :0.161
 3rd Qu.:0.019      3rd Qu.: 0.108   3rd Qu.:0.094   3rd Qu.:0.178
 Max.   :0.044      Max.   : 0.231   Max.   :0.174   Max.   :0.260
```

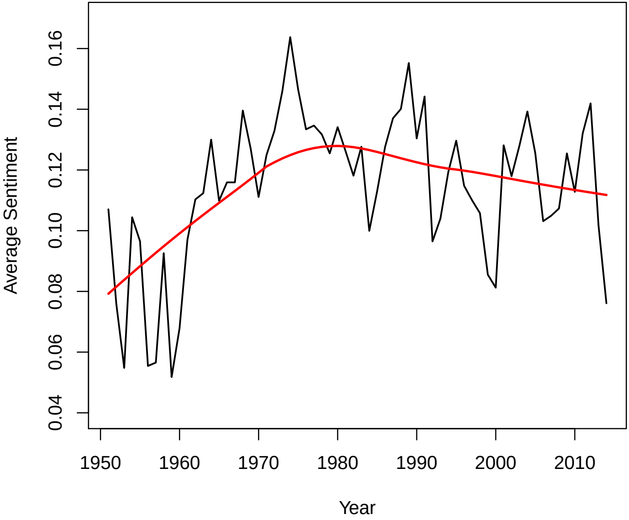
UNHCR: Sentiment vs. Word Count



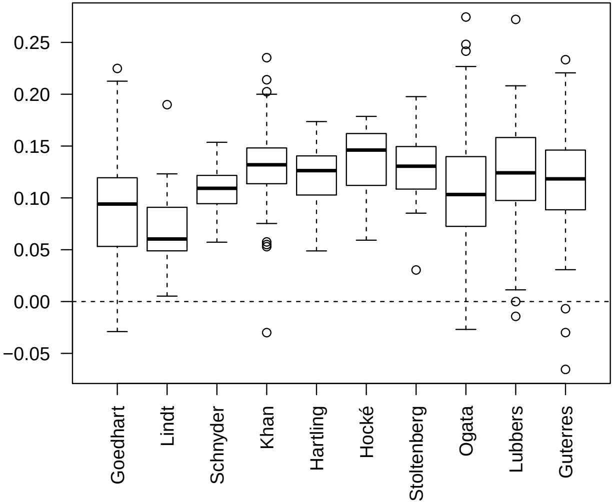
UNHCR: Sentiment Over Time



UNHCR: Annual Sentiment Means



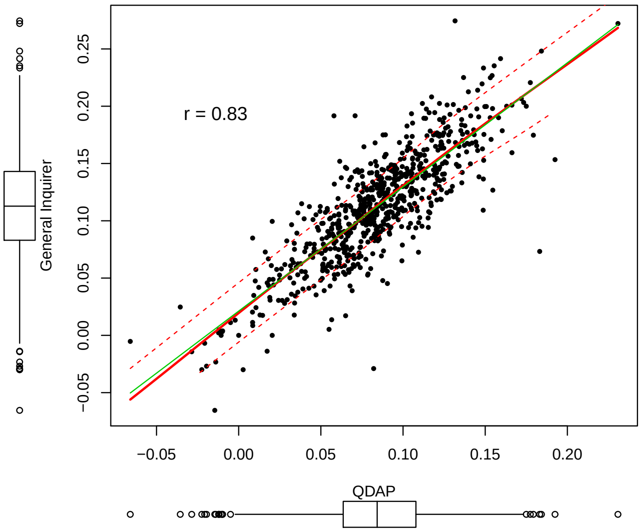
UNHCR: Sentiment By Speaker



Similar Results By Dictionary?

```
> GI<-loadDictionaryGI()
> QD<-loadDictionaryQDAP()
>
> compareDictionaries(GI,QD)
Comparing: binary vs binary

Total unique words: 5100
Matching entries: 2136 (0.42%)
Entries with same classification: 1448 (0.28%)
Entries with different classification: 63 (0.012%)
$totalUniqueWords
[1] 5100

$totalSameWords
[1] 2136

$ratioSameWords
[1] 0.42

$numWordsEqualClass
[1] 1448

$numWordsDifferentClass
[1] 63

$ratioWordsEqualClass
[1] 0.28

$ratioWordsDifferentClass
[1] 0.012
```

Comparing Results w/Different Dictionaries



For Whom Does Dictionary Choice Matter?

```
> DictDiff <- with(UNSent, abs(SentimentGI - SentimentQDAP))

> summary(lm(DictDiff~UN$Author - 1))

Call:
lm(formula = DictDiff ~ UN$Author - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03758 -0.01658 -0.00173  0.01332  0.10766

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
UN$AuthorGoedhart     0.03151    0.00427    7.39  4.4e-13 ***
UN$AuthorLindt        0.01661    0.00444    3.74   0.0002 ***
UN$AuthorSchnyder     0.02043    0.00340    6.01  3.0e-09 ***
UN$AuthorKhan         0.03012    0.00266   11.33  < 2e-16 ***
UN$AuthorHartling     0.03187    0.00296   10.76  < 2e-16 ***
UN$AuthorHocke        0.04397    0.00487    9.04  < 2e-16 ***
UN$AuthorStoltenberg  0.03973    0.00582    6.83  1.8e-11 ***
UN$AuthorOgata        0.03519    0.00133   26.53  < 2e-16 ***
UN$AuthorLubbers      0.03097    0.00255   12.16  < 2e-16 ***
UN$AuthorGuterres     0.03214    0.00203   15.84  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.022 on 693 degrees of freedom
Multiple R-squared:  0.695,Adjusted R-squared:  0.691
F-statistic:  158 on 10 and 693 DF,  p-value: <2e-16
```

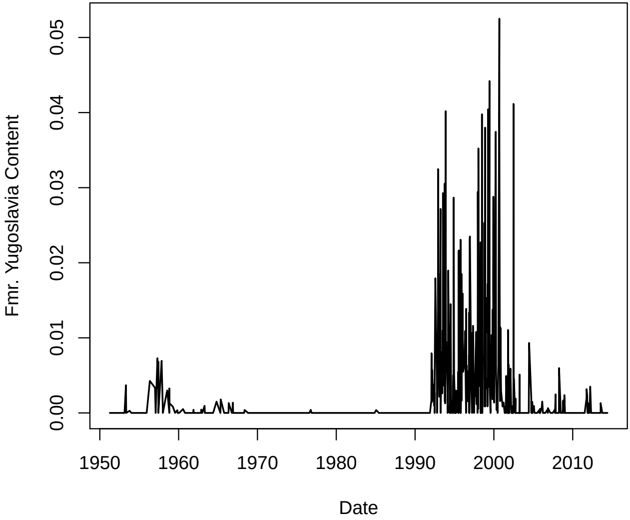
Custom Dictionaries "By Hand"

- Conflict in the former Yugoslavia, 1991-1999
- $\approx$ 2.3 million refugees
- "Europe's biggest refugee crisis since World War II"
- Machine code speeches for content about the former Yugoslavia...

Create and Use a Custom Dictionary

```
> YugoWords <- c("yugoslavia","serbia","bosnia","herzegovina",
+                "kosovo","montenegro","macedonia","croatia",
+                "vojvodina","balkans")

> FmrYugo <- SentimentDictionaryWordlist(YugoWords)

> UNHCRYugo <- analyzeSentiment(UNC,
+                   rules=list("YugoTopic"=list(
+                     ruleRatio,FmrYugo)))

> summary(UNHCRYugo$YugoTopic)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   0.003   0.003   0.053
```

“Former Yugoslavia” Scores Over Time



Other Things We Can Do

- Weight terms in the dictionary
- Generate dictionaries from text (e.g., for author identification)
- **Validate**.

What Should We Actually Do?

Best practice:

- Create dictionary...
- Score a training set of text...
- **Validate!**
  - Assess predictive validity on a test set of text
  - OR: Cross-validate...
  - Compare to human coding / classification!
- Especially important when context matters...

Example: Loughran & McDonald (2011)

- The Harvard IV dictionary assigns negative valence to words that are not negative in accounting/finance (`tax`, `cost`, etc.)
- Also does the reverse (e.g., `litigation`, `misstate`, etc.)

Wrap-Up: Extensions / Challenges / etc.

- **Linguistic complexity**
  - Irony, sarcasm, tone, etc.
  - Complex / subtle negation ("I don't have one guitar; I have many.")
- **Dictionaries**...
  - Specialized vocabularies → standard sentiment dictionaries break down (e.g., "love" in tennis)
  - *Minimally-supervised* dictionary creation (Rice & Zorn)
  - Bleeding edge: *Unsupervised* dictionary creation via negations...
- **Change over time**
  - Word meanings...
  - Word usage...

Topic Models

Topics in Text

- "Topics" / "themes" / etc.: What the document is about.
- How do we know?
  - Word meanings...
  - Clustering of words
  - Tone (sometimes)
- Complications / challenges...
  - What's a "topic"?
  - (Key)words can be ambiguous ("tennis" vs. "crane")
  - Documents are often about > one topic

Extracting Topics

Dictionary-based / Supervised methods

- A la sentiment analysis...
- Predetermined "topics" (think: dictionaries of keywords)
- Topic$_i$ $\rightsquigarrow$ whatever topic(s) have (proportionally) the most terms

Unsupervised methods

- Extract topics from the corpus itself
- Intuition: *co-occurrence* of terms in documents
- Useful when (a) we don't know topics *a priori*, and/or (b) term meaning/usage is complex / nonstandard

Latent Dirichlet Allocation

Intuition:

- Start with N documents $i \in \{1...N\}$ in a corpus
  - Each document i has M_i total words
  - The total of all words in the corpus is V
- Each document comprises a mixture of one or more of k topics
- Each topic comprises a mixture of terms
- We observe documents and terms, but not topics; topics are *latent*
- Goals:
  - Infer the latent topic structure of the corpus
  - Assign documents (probabilistically) to topics
- Process:
  - Assign words to topics
  - Assess Pr(topic | document) and Pr(word | topic)
  - Reassign words to topic
  - Repeat...

LDA: Details

Things:

- Estimation via variational EM or Bayes (Gibbs sampling)
- Result: Vectors of probabilities that document i is in topic k

Choosing K:

- Typically try different values of K
- Choose on the basis of model fit, etc.

Correlated Topic Model (CTM):

- LDA assumes / requires negative covariance between topics
- The **Logistic Normal Distribution** permits some positive covariance between topics...

Structural Topic Models (Roberts et al.)

Intuition: A CTM where topic prevalence (how much of a document is associated with a topic) and/or content (which words go with which topics) varies as a function of document-level metadata predictors.

Some details:

- Predictors enter the MVN via $\boldsymbol{\mu} = \mathbf{Z}_i \gamma$
- No predictors $\equiv$ CTM
- Selection of K is similar to LDA/CTM

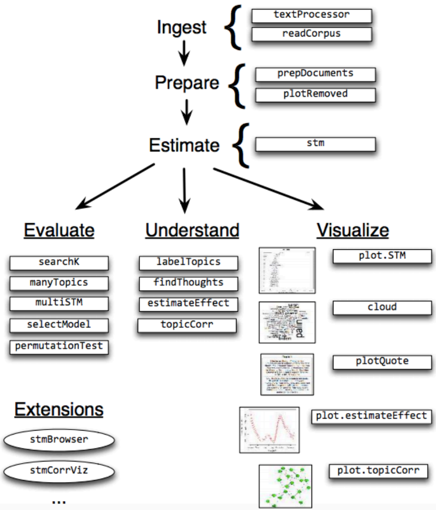
Topic Models in R

- `topicmodels` package
  - Plays well with `tm`
  - LDA and CTM estimation via VEM or Gibbs sampling
  - Some nice graphical tools
  - Is tidy-compatible (see here)
- `stm` package: Structural Topic Models
  - Fits the model in Roberts et al.
  - See the vignette / website
- Others (`quanteda`, `lda`, `text2vec`, `mscstexta4r`)

`topicmodels` Package

- Estimates LDA and CTMs, either via variational approximation (`VEM`, the default) or collapsed Gibbs sampling (`Gibbs`)
- Workhorse functions are `LDA` and `CTM`. Options:
  - `seed` (for replicability)
  - `best` (if `TRUE` (the default), model returns only the model with the highest log-likelihood)
  - Other options related to (VEM or MCMC) optimization...
- Other useful functions:
  - `topics` (extracts most likely topics for each document)
  - `terms` (extracts most likely terms per topic)
  - `posterior` (generates posterior topic probabilities for in- or out-of-sample documents)
  - `perplexity` (calculates model-based perplexity for in- or out-of-sample documents)

stm Package: Example Workflow



(from the vignette)

Example, Redux: UNHCR Speeches

- All speeches made by the High Commissioner of the U.N. Refugee Agency, 1970-2016 ($N = 703$)
- Metadata include ID, speaker, title, and date
- Source: `https://www.kaggle.com/franciscadias/un-refugee-speech-analysis/`

UNHCR Data Prep, Etc.

```
> # Process text (using textProcessor from stm):
> #
> # Note that defaults convert cases, remove stopwords /
> # punctuation / words < 3 characters / extra white space,
> # and stems.
>
> UNHCR <- textProcessor(UN$content, metadata=UN)
Building corpus...
Converting to Lower Case...
Removing punctuation...
Removing stopwords...
Removing numbers...
Stemming...
Creating Output...

> # Create stm corpus. Note that this defaults to dropping
> # words that only appear in one document:
>
> UNCorp <- prepDocuments(UNHCR$documents,UNHCR$vocab,UNHCR$meta)
Removing 6671 of 15742 terms (6671 of 403425 tokens) due to frequency
Your corpus now has 703 documents, 9071 terms and 396754 tokens.>
```

Fit a Standard LDA

```
> UN.LDAV.6 <- LDA(UNLDACorp,6,method="VEM"
+                ,seed=7222009)
> str(UN.LDAV.6)
Formal class 'LDA_VEM' [package "topicmodels"] with 14 slots
  ..@ alpha          : num 0.113
  ..@ call           : language LDA(x = UNLDACorp, k = 6, method = "VEM", seed = 7222009)
  ..@ Dim            : int [1:2] 703 9071
  ..@ control        :Formal class 'LDA_VEMcontrol' [package "topicmodels"] with 13 slots
  .. .. ..@ estimate.alpha: logi TRUE
  .. .. ..@ alpha         : num 8.33
  .. .. ..@ seed          : int 1522857723
  .. .. ..@ verbose       : int 0
  .. .. ..@ prefix        : chr "/var/folders/4p/wkcn3bqs67761813tx05lh9hkvk9km/T//Rtmp8HCEFc/fileba2821eaaa46"
  .. .. ..@ save          : int 0
  .. .. ..@ nstart        : int 1
  .. .. ..@ best          : logi TRUE
  .. .. ..@ keep          : int 0
  .. .. ..@ estimate.beta : logi TRUE
  .. .. ..@ var           :Formal class 'OPTcontrol' [package "topicmodels"] with 2 slots
  .. .. .. .. ..@ iter.max: int 500
  .. .. .. .. ..@ tol     : num 0.000001
  .. .. ..@ em            :Formal class 'OPTcontrol' [package "topicmodels"] with 2 slots
  .. .. .. .. ..@ iter.max: int 1000
  .. .. .. .. ..@ tol     : num 0.0001
  .. .. ..@ initialize    : chr "random"
  ..@ k              : int 6
  ..@ terms          : chr [1:9071] "--camp" "--cuff" "--date" "--job" ...
  ..@ documents      : NULL
  ..@ beta           : num [1:6, 1:9071] -9.34 -225.91 -11.5 -40.89 -26.32 ...
  ..@ gamma          : num [1:703, 1:6] 0.0000786 0.000231 0.0819796 0.0750326 0.0768223 ...
  ..@ wordassignments:List of 5
  .. ..$ i   : int [1:396754] 1 1 1 1 1 1 1 1 1 1 ...
  .. ..$ j   : int [1:396754] 8 48 73 85 107 117 154 174 194 200 ...
  .. ..$ v   : num [1:396754] 6 3 3 6 6 6 6 6 5 6 ...
  .. ..$ nrow: int 703
  .. ..$ ncol: int 9071
  .. ..- attr(*, "class")= chr "simple_triplet_matrix"
  ..@ loglikelihood  : num [1:703] -10851 -3439 -5105 -3402 -4913 ...
  ..@ iter           : int 22
  ..@ logLiks        : num(0)
  ..@ n              : int 906095
```

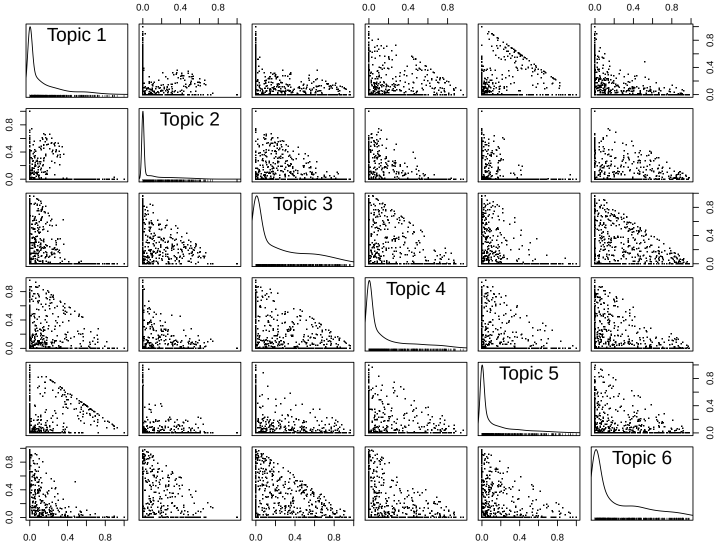
Check Out The Topics

```
> get_terms(UN.LDAV.6,10)

      Topic 1    Topic 2        Topic 3   Topic 4    Topic 5      Topic 6
 [1,] "refuge"   "humanitarian" "refuge"  "unhcr"    "refuge"     "refuge"
 [2,] "countri"  "return"       "unhcr"   "refuge"   "problem"    "intern"
 [3,] "programm" "secur"        "will"    "programm" "work"       "countri"
 [4,] "assist"   "conflict"     "protect" "will"     "nation"     "protect"
 [5,] "govern"   "peac"         "need"    "committe" "commission" "right"
 [6,] "offic"    "displac"      "intern"  "year"     "offic"      "human"
 [7,] "will"     "intern"       "peopl"   "offic"    "high"       "asylum"
 [8,] "problem"  "polit"        "displac" "assist"   "year"       "peopl"
 [9,] "also"     "bosnia"       "countri" "govern"   "unit"       "state"
[10,] "camp"     "forc"         "year"    "continu"  "will"       "nation"
```

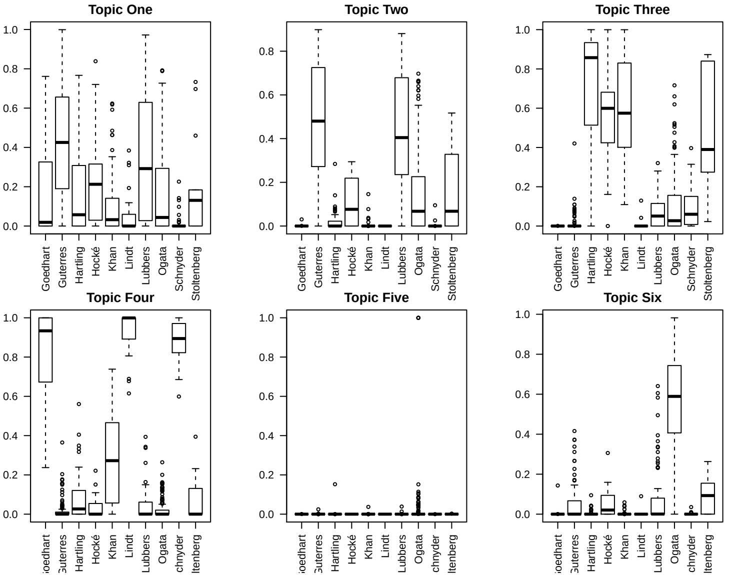
Estimated Pr(Topic | Document)

```
> # Generate posterior probabilities of the topics
> # for each document and the terms for each topic:
>
> V.6.Post <- posterior(UN.LDAV.6)
> cor(V.6.Post$topics)
       1      2      3      4     5      6
1  1.000 -0.138 -0.370 -0.055  0.22 -0.411
2 -0.138  1.000 -0.089 -0.207 -0.24 -0.076
3 -0.370 -0.089  1.000 -0.227 -0.37 -0.189
4 -0.055 -0.207 -0.227  1.000 -0.18 -0.307
5  0.222 -0.245 -0.369 -0.182  1.00 -0.260
6 -0.411 -0.076 -0.189 -0.307 -0.26  1.000
```

Posterior Topic Probabilities



Topic Probabilities by Author



Things to Think About: How Many Topics?

From the `stm` documentation:

> *"The most important user input in parametric topic models is the number of topics. There is no right answer to the appropriate number of topics.* **More topics will give more fine-grained representations of the data at the potential cost of being less precisely estimated.** *The number must be at least 2 which is equivalent to a unidimensional scaling model. For short corpora focused on very specific subject matter (such as survey experiments) 3-10 topics is a useful starting range. For small corpora (a few hundred to a few thousand) 5-50 topics is a good place to start. Beyond these rough guidelines it is application specific. Previous applications in political science with medium sized corpora (10k to 100k documents) have found 60-100 topics to work well. For larger corpora 100 topics is a useful default size. Of course, your mileage may vary." (emphasis added)*

More Things...

- STM integrates measurement and model fitting...
- For STM: Covariates $\rightarrow$ topic *prevalence* or topical *content*?
  - MC region $\rightarrow$ (e.g.) more likely to discuss agriculture, less mass transit
  - MC ideology $\rightarrow$ talk about foreign policy as "humanitarian" vs. "nuclear threat"
- As always, validation is useful...

Text Scaling

Scaling Text

Scaling, so far:

- UDS / MDS, FA/PCA, IRT
- Goal: Combine/aggregate information (data reduction)

Scaling text: Underlying assumptions...

- Individuals speaking/writing/etc. differ in systematic, measurable ways
- Those differences manifest themselves in text...
  - What they say
  - When they say it (topic selection)
  - How they say it (style, tone, etc.)
- The mapping from latent differences to text is *systematic* and observable, and
- Can be learned via analysis of the text itself

Scaling Text (continued)

IRT-type data:

```
> irt.df[1:3,1:7]
       Q1  Q2  Q3  Q4  Q5  Q6  Q7
R1      1   1   1   0   0   0   0
R2      0   0   1   1   1   0   1
R3      1   0   1   0   0   1   0
```

Intuition: Go from binary "correct / incorrect" responses to measures of latent phenomena.

A TDM:

```
> tdm.df[1:3,1:7]
                 ability able about above abroad absolutely abused
Debate2016-1.txt       2   13    78     1      3          5      1
Debate2016-2.txt       0   16   102     1      0          5      0
Debate2016-3.txt       0    6    75     0      0          5      0
```

Intuition: Go from word frequencies / co-occurrences to measures of latent phenomena.

Supervised Text Scoring

Basic idea:

1. We know some documents' / authors' locations
2. Assess which terms in those documents give it it's location (distinctive)
3. Use the resulting term-level scores to locate other documents

One example: "Wordscores" (originally for scoring legislative text: speeches, press releases, etc.)

Wordscores: Things to Remember

- Document scores are (weighted) averages of the words in them, where
- ...the weighting is "according to the proportion of tokens of each word type in the reference document" (Lowe 2008, 357)
- So, words' importance <u>are a function of their frequency</u> in each document type.
- Word-level scores are similar...
- Estimated document scores have vastly underestimated variability
- Issues with rescaling original texts for comparability (Martin and Vanberg 2007; Benoit and Laver 2007)
- Lowe (2008): Wordscores $\leftrightarrow$ Correspondence Analysis $\leftrightarrow$ IRT

Unsupervised Text Scoring

Basic idea:

1. Assume that words $\mathbf{X}$ are generated according to some PDF $f(\cdot)$, with (latent) parameters θ for the units being scaled
2. Assess $\Pr(\theta | f(\cdot), \mathbf{X})$
3. Resulting posterior $\hat{\theta}$ are your scale scores

Characteristics:

- IRT-like...
- One example: "Wordfish" (Slapin and Prokschk 2008) (also originally for scoring legislators)

WordFish...

- Yields estimates of the parameters $(\hat{\theta}, \hat{\alpha}, \hat{\psi}, \hat{\beta})$
- Also provides estimates of variability (method varies by estimation approach)
- More recently: Also estimates ideological clarity / ambiguity (Lo, Proksch and Slapin 2014 *BJPS*)

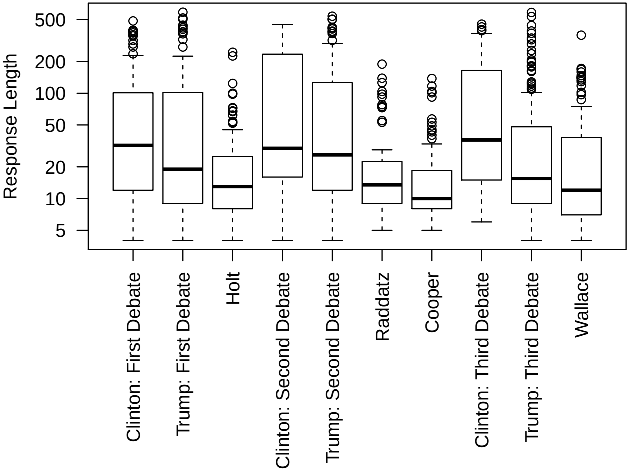
Text Scaling: Options in R

- `quanteda` (Benoit et al.)
- `austin` (Lowe)
- Various others (e.g., Slapin's `wordfish` code)

Example: The 2016 Presidential Debates

- Transcripts from all three general election (Clinton/Trump) debates
  - First Debate: 9/26/16, Hofstra University (Lester Holt moderating)
  - Second Debate: 10/9/16, Washington University (Martha Raddatz and Anderson Cooper moderating, town hall format)
  - Third Debate: 10/19/16, UNLV (Chris Wallace moderating)
- $N = 922$ "documents" (instances of one person speaking), 3986 sentences, 59256 tokens (34943 unique terms)
- Goals:
  - Scale Clinton, Trump, perhaps the moderators
  - Assess change from one debate to the next
  - ???

Length of Responses



Diversion: "Keyness"

Q: How good is a word (say, "terrorist") at *discriminating* among documents?

- Equally common (or rare) in both = not very
- Common in one, rare in the other = *very*

Intuition: a χ^2 statistic from a 2×2 frequency table:

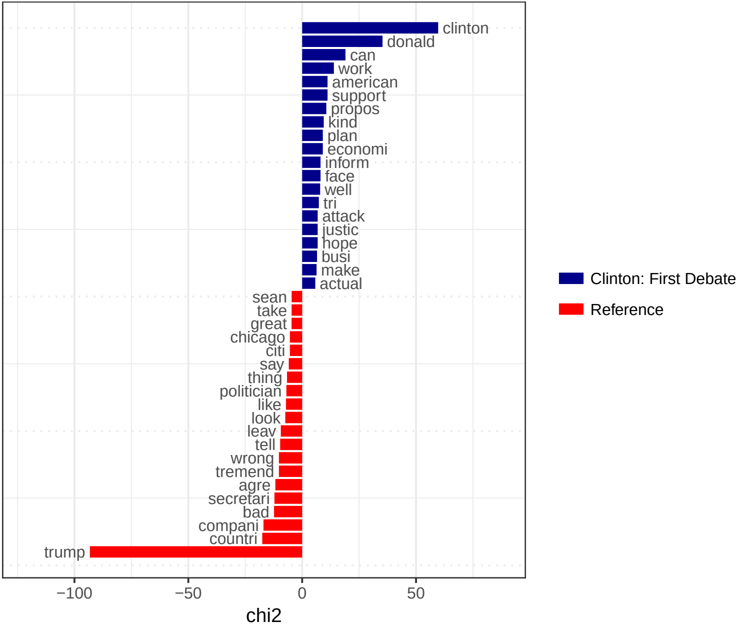
	"terrorist"	All other words	Total
Document A	N_{TA}	N_{OA}	N_A
Document B	N_{TB}	N_{OB}	N_B
Total	N_T	N_A	N

Larger values of $\chi^2 \rightarrow$ higher "keyness"

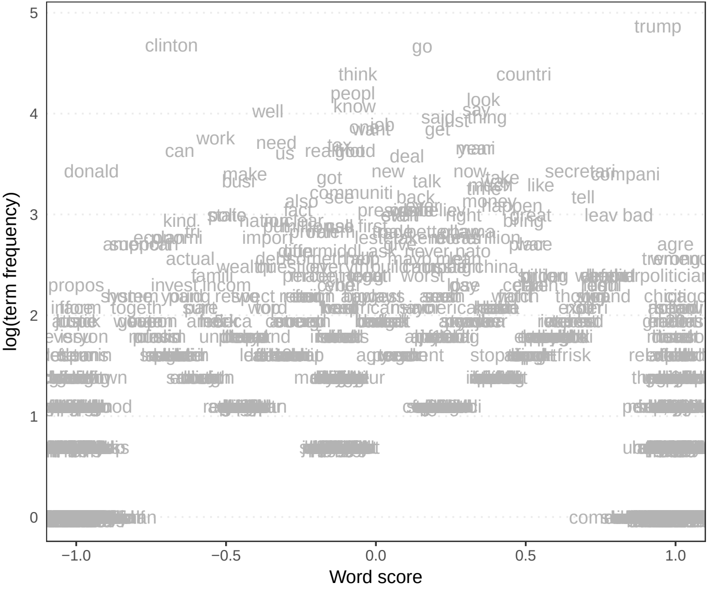
Word "Keyness": First Debate

```
> D1C2<-corpus_subset(D1C,Speaker %in% c("Clinton: First Debate",
+                                          "Trump: First Debate"))
> D1C2DFM <- dfm(D1C2,remove=stopwords("english"),stem=TRUE,
+            remove_punct=TRUE,groups="Speaker")
>
> D1Key <- textstat_keyness(D1C2DFM, target = "Clinton: First Debate")
>
> head(D1Key,12)
    feature   chi2         p n_target n_reference
1   clinton 59.722 1.088e-14       87          21
2    donald 35.291 2.840e-09       30           1
3       can 19.012 1.299e-05       30           8
4      work 13.924 1.903e-04       31          12
5   support 11.142 8.440e-04       13           2
6  american 11.142 8.440e-04       13           2
7    propos 10.600 1.131e-03       10           0
8      kind  9.480 2.078e-03       15           4
9   economi  9.064 2.607e-03       13           3
10     plan  9.064 2.607e-03       13           3
11     face  8.068 4.506e-03        8           0
12   inform  8.068 4.506e-03        8           0
```

First Debate Keyness Differentials

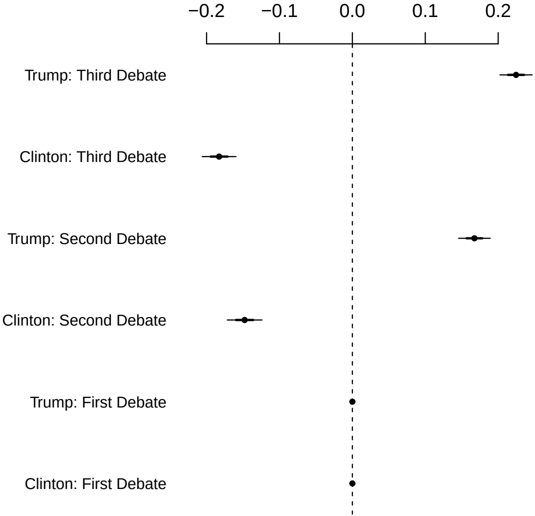


Word Scores...



log(term frequency)
Word score
5
4
3
2
1
0
−1.0
−0.5
0.0
0.5
1.0
trump
clinton
go
think
countri
peopl
look
know
well
say
said
thing
get
work
need
can
us
tax
deal
year
donald
make
busi
new
now
take
secretari
compani
got
talk
like
communiti
also
see
back
money
tell
fact
happen
state
nuclear
great
leav
bad
kind
right
bring
first
obama
import
million
place
nato
agre
actual
china
famili
worst
politician
propos
cyber
invest
incom
chicago
togeth
america
stop
frisk

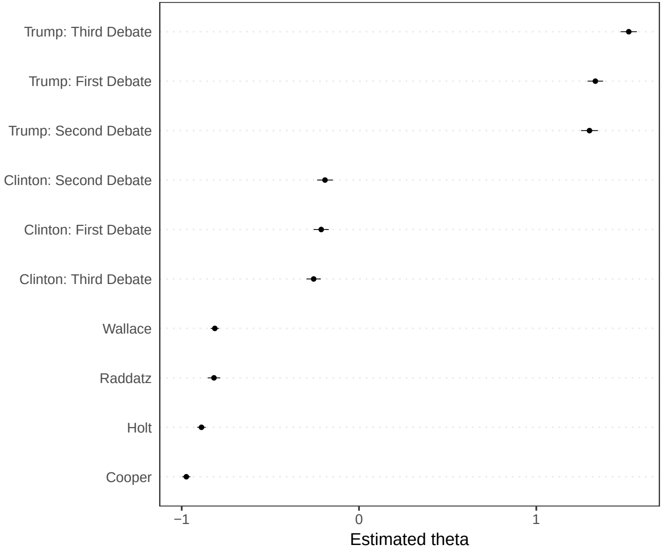
Wordscores: Ladder Plot



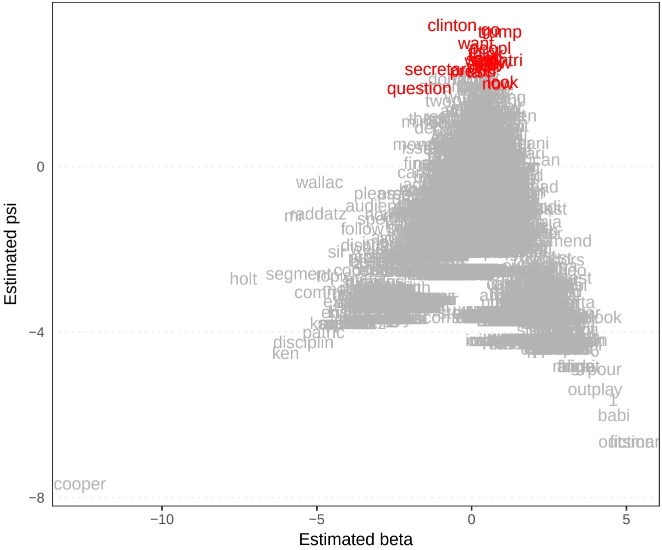
Wordfish!

```
> WF <- textmodel_wordfish(DDFM,dir=c(1,2))
> summary(WF)

Call:
textmodel_wordfish.dfm(x = DDFM, dir = c(1, 2))

Estimated Document Positions:
                        theta     se
Clinton: First Debate  -0.213 0.0218
Trump: First Debate     1.333 0.0224
Holt                   -0.889 0.0123
Clinton: Second Debate -0.192 0.0227
Trump: Second Debate    1.300 0.0243
Raddatz                -0.818 0.0177
Cooper                 -0.974 0.0115
Clinton: Third Debate  -0.256 0.0204
Trump: Third Debate     1.522 0.0233
Wallace                -0.813 0.0119

Estimated Feature Scores:
     clinton donald applaus  well thank lester hofstra   host    us central
beta  -0.626 -0.579  0.0833 0.252 -1.36  0.606   -1.43 -0.578 0.211   -1.15
psi    3.425  2.119  0.5430 2.571  1.17  0.139   -2.13 -1.427 1.948   -1.61
     question  elect realli   kind countri  want  futur build togeth  today
beta    -1.69 -0.513   0.57 -0.147   0.821 0.145 -0.561 0.679 -0.375 0.4548
psi      1.90  1.225   1.72  1.326   2.583 2.992 -0.442 0.441  0.553 0.0914
     granddaught second birthday think   lot   first economi    work
beta      -0.451  0.214   -0.451  0.46 0.383 -0.0648   -0.25 -0.0746
psi       -2.502  1.159   -2.502  2.75 2.077  1.9229    1.01  2.1613
     everyon  just
beta  -0.267 0.504
psi    0.351 2.478
```

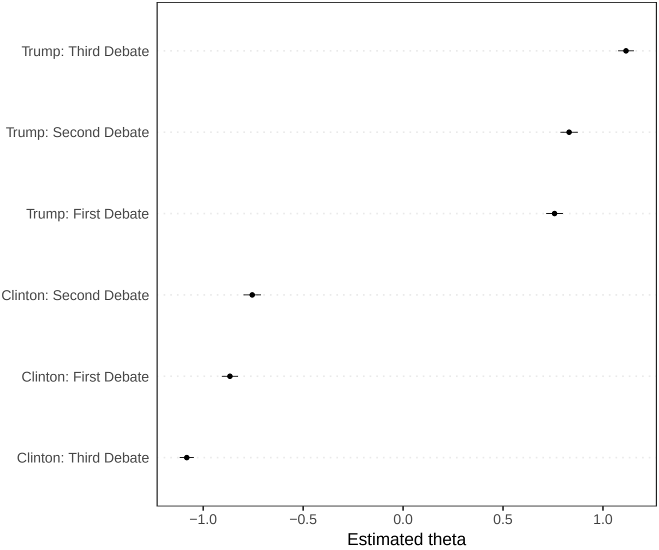
Wordfish: Speaker Locations



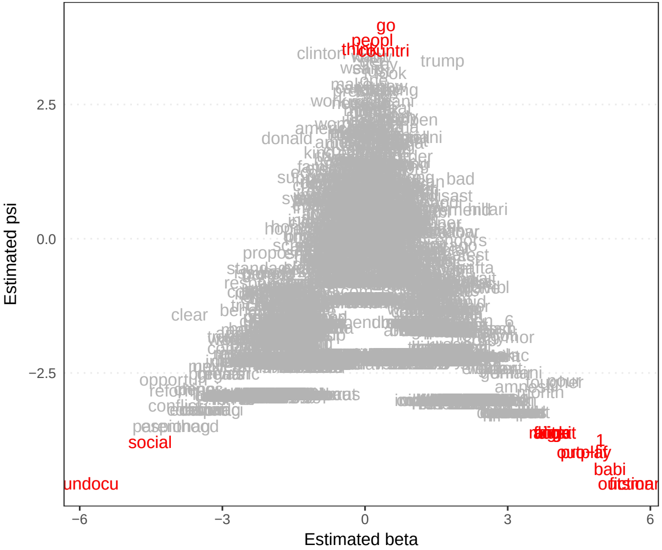
Wordfish: Word Locations



Wordfish: Speaker Locations (Candidates Only)



Wordfish: Word Locations (large $|\hat{\psi}|$)



Scaling Texts: Other Approaches...

- E.g., factor analysis / SEM, unfolding, IRT...
- The "Class Affinity Model"
    - Perry and Benoit (2017)
    - "...a text modeling framework that allows actors to take latent positions on a 'gray' spectrum between 'black' and 'white' polar opposites."
    - In `quanteda`
- They're all kinda the same.
- (Read this paper by Will Lowe: `http://dl.conjugateprior.org/preprints/all-on-the-line.pdf`)

Scaling Texts: Things to Think About

- Interpretation: What do the scales mean?

- What does it mean to "validate"?
  - Compare to human / expert coding?
  - Compare to "numerical" position estimates (D-NOMINATE / Martin-Quinn / etc.)?
  - Cross-validate?
  - Predicting other phenomena

- Propagating (measurement and estimation) uncertainty...

Thank you.