CEU CENTRAL EUROPEAN UNIVERSITY

VU

Automated Text Analysis in Political Science

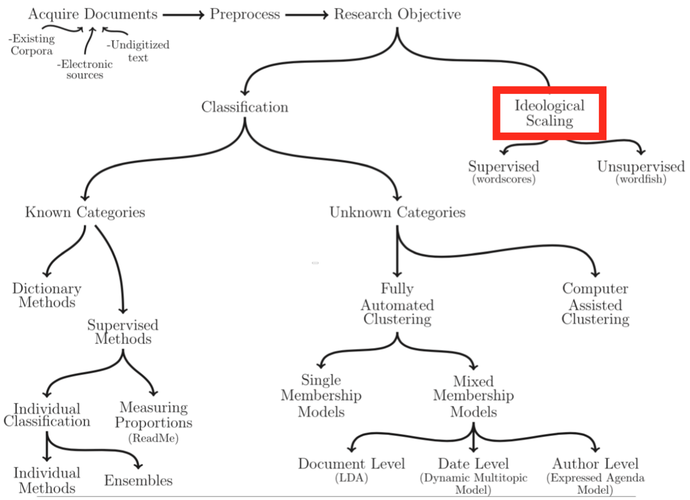
Martijn Schoonvelde

23 April 2018

Today's class

- Use text to study party positions
  - Central aim for political scientists, dating back to first spatial models (Downs, 1957)
- Two approaches:
  - Wordscores (supervised method)
  - Wordfish (unsupervised method)
- Validation and reliability; when are these methods retrieving an ideological position?
- Practice using R

Overview of Text as Data Methods



(Grimmer and Stewart, 2013, 268)

Scaling methods

- Developed to study (latent) party positions
- Traditionally measured using expert surveys (CHES) and hand coding (CMP) or roll call votes
  - Expensive and labor intensive, and not always available
    - For example, CMP measures policy positions by coding and aggregating every ‘quasi sentence‘ in a manifesto to a policy area
  - Also: much debate over how to aggregate across policy areas into ideological dimensions
- Two types of scaling methods:
  - Supervised approach: Wordscores (Laver, Benoit and Garry, 2003)
  - Unsupervised approach: Wordfish (Slapin and Proksch, 2008)

Scaling methods

- Identifying assumption: *ideological dominance* (Grimmer and Stewart 2013)
- Rate at which politicians use certain words depends on their ideological position
- Maybe true for some texts but not for others – requires careful validation

Wordscores

- Procedure goes as follows:
  - Identify ‘reference texts‘ that represent the extremes of the political space (say, left-right)
    - Reference values assigned to these texts, ideally from (independent) expert surveys
  - Each word in the reference text is assigned a score (Wordscore)
  - Relative rate each word is used in the reference texts
  - Wordscores used to scale remaining texts – placed in between the reference texts

Calculating Wordscores

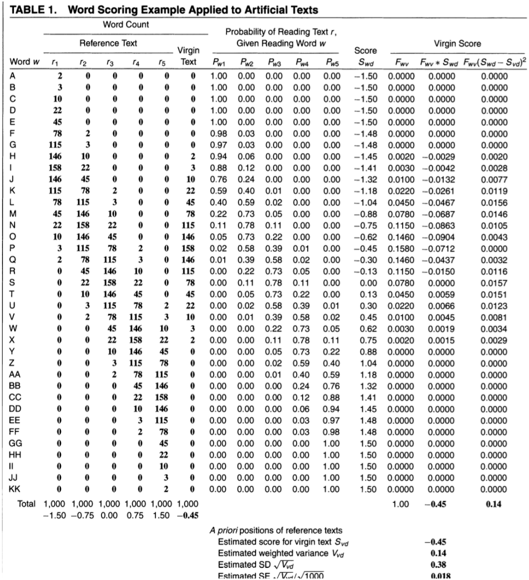


TABLE 1. Word Scoring Example Applied to Artificial Texts

	Word Count						Probability of Reading Text r, Given Reading Word w						Virgin Score		
	Reference Text					Virgin						Score			
Word w	r_1	r_2	r_3	r_4	r_5	Text	P_{w1}	P_{w2}	P_{w3}	P_{w4}	P_{w5}	S_{wd}	F_{wv}	$F_{wv} * S_{wd}$	$F_{wv}(S_{wd} - S_{vd})^2$
A	2	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	0.0000	0.0000	0.0000
B	3	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	0.0000	0.0000	0.0000
C	10	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	0.0000	0.0000	0.0000
D	22	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	0.0000	0.0000	0.0000
E	45	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	0.0000	0.0000	0.0000
F	78	2	0	0	0	0	0.98	0.03	0.00	0.00	0.00	−1.48	0.0000	0.0000	0.0000
G	115	3	0	0	0	0	0.97	0.03	0.00	0.00	0.00	−1.48	0.0000	0.0000	0.0000
H	146	10	0	0	0	2	0.94	0.06	0.00	0.00	0.00	−1.45	0.0020	−0.0029	0.0020
I	158	22	0	0	0	3	0.88	0.12	0.00	0.00	0.00	−1.41	0.0030	−0.0042	0.0028
J	146	45	0	0	0	10	0.76	0.24	0.00	0.00	0.00	−1.32	0.0100	−0.0132	0.0077
K	115	78	2	0	0	22	0.59	0.40	0.01	0.00	0.00	−1.18	0.0220	−0.0261	0.0119
L	78	115	3	0	0	45	0.40	0.59	0.02	0.00	0.00	−1.04	0.0450	−0.0467	0.0156
M	45	146	10	0	0	78	0.22	0.73	0.05	0.00	0.00	−0.88	0.0780	−0.0687	0.0146
N	22	158	22	0	0	115	0.11	0.78	0.11	0.00	0.00	−0.75	0.1150	−0.0863	0.0105
O	10	146	45	0	0	146	0.05	0.73	0.22	0.00	0.00	−0.62	0.1460	−0.0904	0.0043
P	3	115	78	2	0	158	0.02	0.58	0.39	0.01	0.00	−0.45	0.1580	−0.0712	0.0000
Q	2	78	115	3	0	146	0.01	0.39	0.58	0.02	0.00	−0.30	0.1460	−0.0437	0.0032
R	0	45	146	10	0	115	0.00	0.22	0.73	0.05	0.00	−0.13	0.1150	−0.0150	0.0116
S	0	22	158	22	0	78	0.00	0.11	0.78	0.11	0.00	0.00	0.0780	0.0000	0.0157
T	0	10	146	45	0	45	0.00	0.05	0.73	0.22	0.00	0.13	0.0450	0.0059	0.0151
U	0	3	115	78	2	22	0.00	0.02	0.58	0.39	0.01	0.30	0.0220	0.0066	0.0123
V	0	2	78	115	3	10	0.00	0.01	0.39	0.58	0.02	0.45	0.0100	0.0045	0.0081
W	0	0	45	146	10	3	0.00	0.00	0.22	0.73	0.05	0.62	0.0030	0.0019	0.0034
X	0	0	22	158	22	2	0.00	0.00	0.11	0.78	0.11	0.75	0.0020	0.0015	0.0029
Y	0	0	10	146	45	0	0.00	0.00	0.05	0.73	0.22	0.88	0.0000	0.0000	0.0000
Z	0	0	3	115	78	0	0.00	0.00	0.02	0.59	0.40	1.04	0.0000	0.0000	0.0000
AA	0	0	2	78	115	0	0.00	0.00	0.01	0.40	0.59	1.18	0.0000	0.0000	0.0000
BB	0	0	0	45	146	0	0.00	0.00	0.00	0.24	0.76	1.32	0.0000	0.0000	0.0000
CC	0	0	0	22	158	0	0.00	0.00	0.00	0.12	0.88	1.41	0.0000	0.0000	0.0000
DD	0	0	0	10	146	0	0.00	0.00	0.00	0.06	0.94	1.45	0.0000	0.0000	0.0000
EE	0	0	0	3	115	0	0.00	0.00	0.00	0.03	0.97	1.48	0.0000	0.0000	0.0000
FF	0	0	0	2	78	0	0.00	0.00	0.00	0.03	0.98	1.48	0.0000	0.0000	0.0000
GG	0	0	0	0	45	0	0.00	0.00	0.00	0.00	1.00	1.50	0.0000	0.0000	0.0000
HH	0	0	0	0	22	0	0.00	0.00	0.00	0.00	1.00	1.50	0.0000	0.0000	0.0000
II	0	0	0	0	10	0	0.00	0.00	0.00	0.00	1.00	1.50	0.0000	0.0000	0.0000
JJ	0	0	0	0	3	0	0.00	0.00	0.00	0.00	1.00	1.50	0.0000	0.0000	0.0000
KK	0	0	0	0	2	0	0.00	0.00	0.00	0.00	1.00	1.50	0.0000	0.0000	0.0000
Total	1,000	1,000	1,000	1,000	1,000	1,000							1.00	**−0.45**	**0.14**
	−1.50	−0.75	0.00	0.75	1.50	**−0.45**									

A priori positions of reference texts

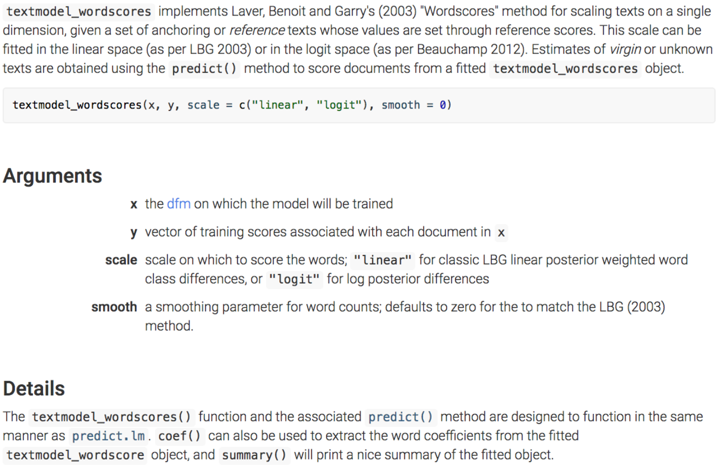
Estimated score for virgin text S_{vd} **−0.45**

Estimated weighted variance V_{vd} **0.14**

Estimated SD $\sqrt{V_{vd}}$ **0.38**

Estimated SE $\sqrt{V_{vd}}/\sqrt{1000}$ **0.018**

Wordscores in Quanteda



`textmodel_wordscores` implements Laver, Benoit and Garry's (2003) "Wordscores" method for scaling texts on a single dimension, given a set of anchoring or *reference* texts whose values are set through reference scores. This scale can be fitted in the linear space (as per LBG 2003) or in the logit space (as per Beauchamp 2012). Estimates of *virgin* or unknown texts are obtained using the `predict()` method to score documents from a fitted `textmodel_wordscores` object.

```
textmodel_wordscores(x, y, scale = c("linear", "logit"), smooth = 0)
```

Arguments

x the dfm on which the model will be trained

y vector of training scores associated with each document in `x`

scale scale on which to score the words; `"linear"` for classic LBG linear posterior weighted word class differences, or `"logit"` for log posterior differences

smooth a smoothing parameter for word counts; defaults to zero for the to match the LBG (2003) method.

Details

The `textmodel_wordscores()` function and the associated `predict()` method are designed to function in the same manner as `predict.lm`. `coef()` can also be used to extract the word coefficients from the fitted `textmodel_wordscore` object, and `summary()` will print a nice summary of the fitted object.

Criticisms of Wordscores

- Success hinges on whether reference texts indeed span the extremes of a political space
- May conflate stylistic differences between politicians with ‘ideological language’
  - Could be solved with careful pre-processing (but remember Denny and Spirling, 2018)
- There is no underlying model of how text is generated (Lower 2008)
- How to deal with change over time? Can reference texts from election t meaningfully constrain manifestos at time $t + 1$?

Wordfish

- Also assumes that word usages is driven by ideology / policy positions
- But no reference texts
  - Unsupervised method
- Word usage to be generated by a Poisson process
  - Poisson is a discrete probability distribution "of the number of events occurring in a given time period, given the average number of times the event occurs over that time period."
  - Parameter λ modeled as a function of word and document characteristics

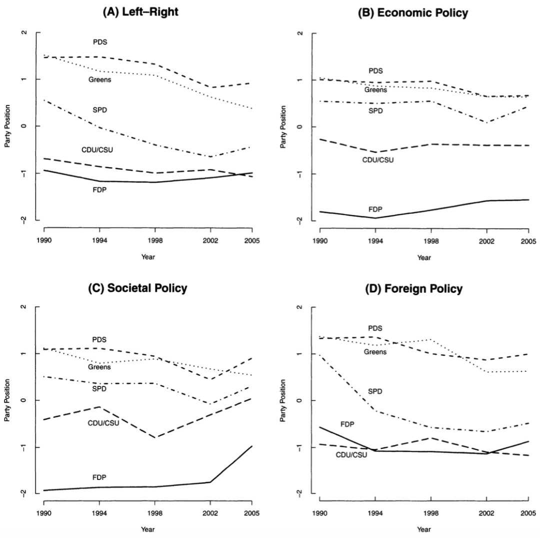
Wordfish model

$$y_{ijt} \sim Poisson(\lambda_{ijt})$$
$$\lambda_{ijt} = exp(\alpha_{it} + \psi_j + \beta_j * \theta_{it})$$

- y_{ijt} is the count of word j in party i's manifesto in year t
- α party-election year fixed effects
  - accounts for the possibility that some parties in some years may have written a much longer manifesto
- ψ word fixed effects
  - accounts for the fact that some words are used more than other words by all parties
- β is a word weight, capturing the importance of word j in discriminating between party positions
- θ the estimate of party i's position in year t (in original paper referred to as ω)

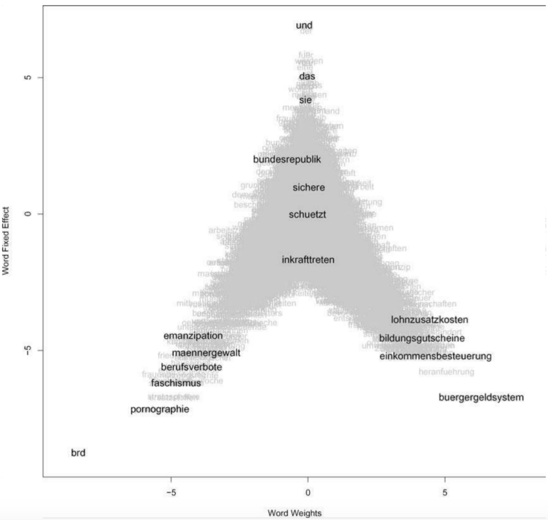
Wordfish model: documents

FIGURE 1 Estimated Party Positions in Germany, 1990–2005



Wordfish model: words

FIGURE 2 Word Weights vs. Word Fixed Effects. Left-Right Dimension, Germany 1990–2005 (Translations given in text)



Validation efforts

- Hjorth et al. (2015) validate Wordscores and Wordfish for German and Danish manifestos

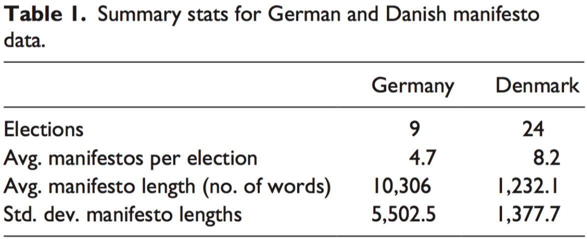


Table 1. Summary stats for German and Danish manifesto data.

	Germany	Denmark
Elections	9	24
Avg. manifestos per election	4.7	8.2
Avg. manifesto length (no. of words)	10,306	1,232.1
Std. dev. manifesto lengths	5,502.5	1,377.7

- Variation in number of parties, manifesto length and historical contingencies
- Validated against:
  - Expert surveys (CHES) and Damgaard (2000)
  - Voter placement of parties (Eurobarometer)
    - Average self-placement of voters who intend to vote for a party
  - CMP RILE measure

Danish results

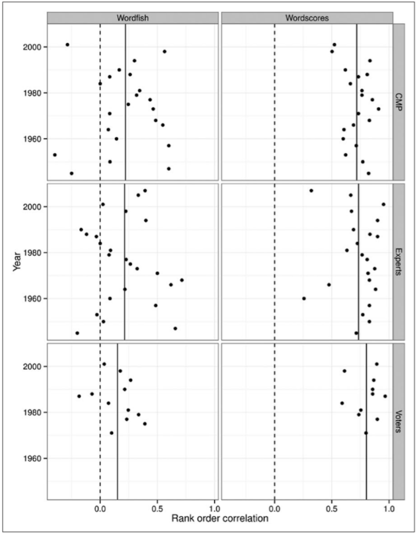


Figure 1. Wordfish and Wordscores estimates' rank order correlations with CMP, expert and voter estimates for each election year in the Danish sample. Vertical lines signify average rank order correlation across years.

German results

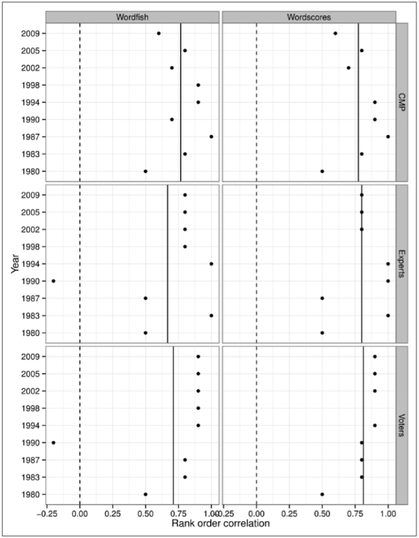


Figure 2. Wordfish and Wordscores estimates' rank order correlations with CMP, expert and voter estimates for each election year in the German sample. Vertical lines signify average rank order correlation across years.

Recommendations Hjorth et al. (2015)

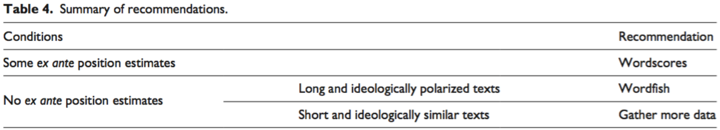
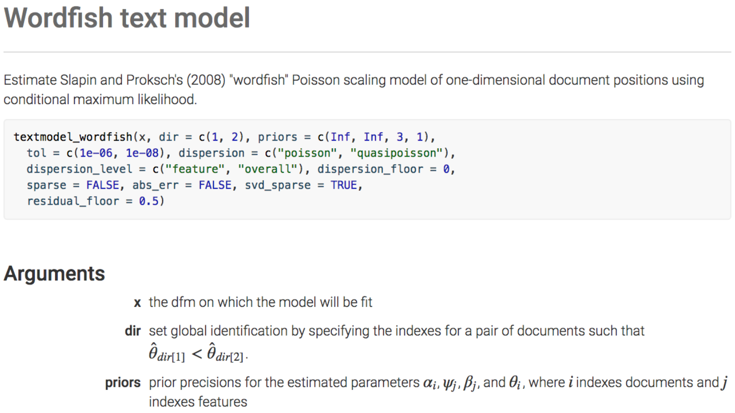


Table 4. Summary of recommendations.

Conditions		Recommendation
Some *ex ante* position estimates		Wordscores
No *ex ante* position estimates	Long and ideologically polarized texts	Wordfish
	Short and ideologically similar texts	Gather more data

Wordfish in Quanteda



Wordfish text model

Estimate Slapin and Proksch's (2008) "wordfish" Poisson scaling model of one-dimensional document positions using conditional maximum likelihood.

```
textmodel_wordfish(x, dir = c(1, 2), priors = c(Inf, Inf, 3, 1),
  tol = c(1e-06, 1e-08), dispersion = c("poisson", "quasipoisson"),
  dispersion_level = c("feature", "overall"), dispersion_floor = 0,
  sparse = FALSE, abs_err = FALSE, svd_sparse = TRUE,
  residual_floor = 0.5)
```

Arguments

x the dfm on which the model will be fit

dir set global identification by specifying the indexes for a pair of documents such that $\hat{\theta}_{dir[1]} < \hat{\theta}_{dir[2]}$.

priors prior precisions for the estimated parameters α_i, ψ_j, β_j, and θ_i, where i indexes documents and j indexes features