



university of
groningen

Quantitative Text Analysis – Essex Summer School

Scaling methods

dr. Martijn Schoonvelde

University of Groningen

Today's class

- Using text to position documents on a single (ideological) dimension
 - Wordscores (supervised approach)
 - Wordfish (unsupervised approach)
 - LSS (semisupervised approach)
- Flash talks: Lisanne, Malika
- Lab session

Research task: position documents / actors along a **single ideological dimension**

- Traditionally measured using **expert surveys** (CHES), **hand coding** (CMP) or **roll call votes**
 - Expensive, labor intensive, and not always available

Idea: use the **text these actors produce** to capture positions

- Supervised approach: Wordscores (Laver, Benoit and Garry, 2003)
- Unsupervised approach: Wordfish (Slapin and Proksch, 2008)
- Semisupervised approach: LSS (Watanabe, 2020)

Scaling methods

- Identifying assumption: **ideological dominance** (Grimmer and Stewart 2013) – rate at which political actors use certain words depends on their ideological position
- For example, **death tax** versus **estate tax** in the United States to describe federal taxes on assets of the deceased

Republicans put a high level of importance on the death/estate tax language – they had to work hard to get members to act in unison, including training members to say ‘death tax’... Estate tax sounds like it only hits the wealthy but ‘death tax’; sounds like it hits everyone (Anonymous Republican staffer. In: Graetz & Shapiro 2005)

- Maybe true for some texts but not for others – **requires careful validation**

Wordscores

The procedure follows these steps:

1. Identify 'reference texts' that represent the **extremes of a political spectrum** (e.g., left-right, liberal-conservative, government-opposition, pro-EU-anti-EU)
2. Assign reference values to these texts.
3. Each word in the reference texts is assigned a **Wordscore**, calculated based on the relative frequency of each word in these texts multiplied by the reference values.
4. Wordscores are then used to scale unseen texts by **summing the product of all Wordscores and their relative frequencies** in these texts.

TABLE 1. Word Scoring Example Applied to Artificial Texts

Word Count									
Reference Text					Probability of Reading Text r_i , Given Reading Word w				
Word w	r_1	r_2	r_3	r_4	r_5	Text	P_{w1}	P_{w2}	P_{w3}
A	2	0	0	0	0	0	1.00	0.00	0.00
B	3	0	0	0	0	0	1.00	0.00	0.00
C	10	0	0	0	0	0	1.00	0.00	0.00
D	22	0	0	0	0	0	1.00	0.00	0.00
E	45	0	0	0	0	0	1.00	0.00	0.00
F	70	2	0	0	0	0	0.98	0.02	0.00
G	115	3	0	0	0	0	0.97	0.03	0.00
H	146	10	0	0	0	2	0.94	0.06	0.00
I	150	22	0	0	3	0	0.88	0.12	0.00
J	146	45	0	0	0	10	0.78	0.24	0.00
K	115	70	2	0	0	22	0.59	0.40	0.01
L	70	115	3	0	0	45	0.40	0.59	0.02
M	45	146	10	0	0	70	0.22	0.75	0.02
N	22	150	22	0	0	115	0.11	0.78	0.11
O	10	146	45	0	0	146	0.05	0.73	0.22
P	3	115	70	2	0	150	0.02	0.58	0.39
Q	2	70	115	3	0	146	0.01	0.39	0.58
R	0	45	146	10	0	115	0.00	0.22	0.73
S	0	22	150	22	0	70	0.00	0.11	0.78
T	0	10	146	45	0	45	0.00	0.05	0.73
U	0	3	115	70	2	22	0.00	0.02	0.58
V	0	2	70	115	3	10	0.00	0.01	0.39
W	0	0	45	146	10	3	0.00	0.00	0.22
X	0	0	22	150	22	2	0.00	0.00	0.11
Y	0	0	10	146	45	0	0.00	0.00	0.05
Z	0	0	3	115	70	0	0.00	0.00	0.02
AA	0	0	0	22	150	0	0.00	0.00	0.01
BB	0	0	0	45	146	0	0.00	0.00	0.00
CC	0	0	0	22	150	0	0.00	0.00	0.00
DD	0	0	0	10	146	0	0.00	0.00	0.00
EE	0	0	0	3	115	0	0.00	0.00	0.00
FF	0	0	0	2	70	0	0.00	0.00	0.00
GG	0	0	0	0	45	0	0.00	0.00	0.00
HH	0	0	0	0	22	0	0.00	0.00	0.00
II	0	0	0	0	10	0	0.00	0.00	0.00
JJ	0	0	0	0	3	0	0.00	0.00	0.00
KK	0	0	0	0	2	0	0.00	0.00	0.00
Total	1,000	1,000	1,000	1,000	1,000	1,000	-1.50	-0.75	0.00
							0.75	1.50	-0.45

A priori positions of reference texts	
Estimated score for virgin text S_{vir}	-0.45
Estimated weighted variance V_{wv}	0.14
Estimated SD $\sqrt{V_{wv}}$	0.38
Estimated SE $\sqrt{V_{wv}/1000}$	0.012

Source: Laver, Benoit & Garry, 2003

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'
- There is no underlying model of how text is generated (Lowe 2008)
 - It's a mechanical operation applied to a document vector
- How to deal with change over time? Can reference texts from election t meaningfully constrain manifestos at time $t + 1$?
- See Bruinsma & Gemenis (2019) for an **extensive critique**

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.

Wordscores in Quanteda

```
library(quanteda); library(quanteda.textmodels)

> dfmat <- tokens(c("socialist worker", "worker europe taxes", "taxes europe", "taxes security
europe")) %>% dfm()
> docvars(dfmat, "reference_score") <- c(-1, NA, NA, 1)
> speeches_ws <- textmodel_wordscores(dfmat,
+                                     y = docvars(dfmat, "reference_score"),
+                                     scale = c("linear"),
+                                     smooth = 0)
> speeches_ws_predict <- predict(speeches_ws, newdata = dfmat)
> speeches_ws_predict
      text1      text2      text3      text4
-1.0000000  0.3333333  1.0000000  1.0000000
> speeches_ws$wordscores
socialist      worker      europe      taxes      security
      -1          -1          1          1          1
```


- Assumes that the relative word usage of parties provides information about their placement in a policy space.
- **Unsupervised method** – there are no reference texts.
- Word usage is generated by a **Poisson process**.
 - The Poisson process is a discrete probability distribution that describes the number of events occurring in a given time period, given the average number of times the event occurs over that time period.
 - The parameter λ is modeled as a function of word and document characteristics.

Wordfish Model: Application

Schwemmer & Wieczorek (2020) use Wordfish to identify **qualitative-quantitative divide** in sociology journals

- Wordfish model applied to 8737 abstracts of articles published in general Sociology journals between 1995 and 2017

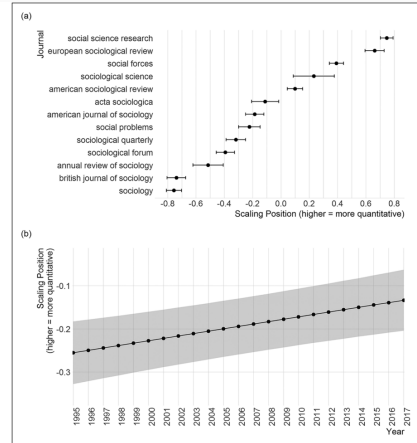


Figure 3. Predicted values for abstract scaling positions by journal (a) and publication year (b) with 95% confidence intervals.

Source: Schwemmer & Wieczorek, 2020

$$y_{ij} \sim \text{Poisson}(\lambda_{ij})$$
$$\lambda_{ij} = \exp(\alpha_i + \psi_j + \beta_j * \theta_i)$$

Applied to the Sociology abstracts, this models the number of times abstract i includes term j .

- Alpha α is a document-level-fixed effect, controlling for some abstracts including more terms than others.
- Psi ψ is word-fixed-effect, controlling for some terms being used more frequently than others.
- Beta β is the estimate weight for each term used to position the documents on the one-dimensional scale.
- Theta θ is the estimate for each document (abstract) on the one-dimensional scale.

The Wordfish model is estimated using an **Expectation Maximization (EM)** algorithm (conditional maximum likelihood) as follows:

1. Calculate starting values for all parameters
2. Estimate party parameters keeping word parameters fixed
3. Estimate word parameters keeping updated document parameters fixed
4. Repeat until convergence

Wordfish Model: documents

Schwemmer & Wieczorek plot the estimated θ for each document over time.

1. Is there a general divide between qualitative and quantitative work in sociology?
2. Is sociology becoming more quantitative?

To answer these questions they **model document scores** across journals and years

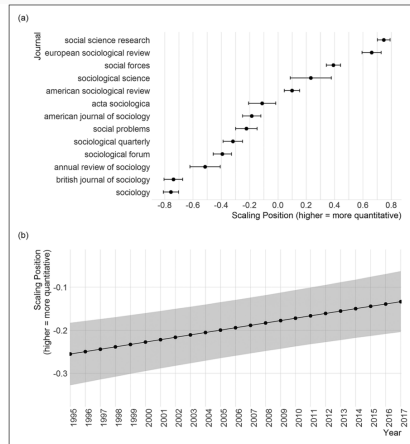


Figure 3. Predicted values for abstract scaling positions by journal (a) and publication year (b) with 95% confidence intervals.

Source: Schwemmer & Wieczorek, 2020

Wordfish model: words

When plotting estimated term weights and word fixed effects against each other this often produces an **eiffel plot**

- Words that are used often but do not help distinguish between quantitative and qualitative abstracts: **high fixed effects and low term weights**
- Words that help distinguish between quantitative and qualitative abstracts and are used less often: **low fixed effects and high term weights** (in either direction)

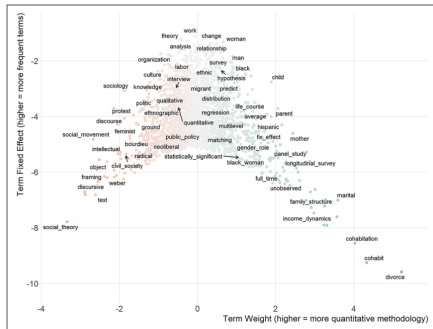


Figure 2. Term weights and fixed effects from the wordfish scaling model fitted to Sociology abstracts.

Source: Schwemmer & Wieczorek, 2020

As always, validation steps are **context-specific**, but they should at least include the following

- Inspecting term plots is a **necessary step** to identify the meaning of the underlying dimension
- **Read documents** that score particularly high on your estimated dimension. Contrast against human coding
- Also: consider Denny & Spirling (2018): to what extent is the estimated underlying dimension **conditional on pre-processing steps**?

Validation efforts

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

Table I. 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

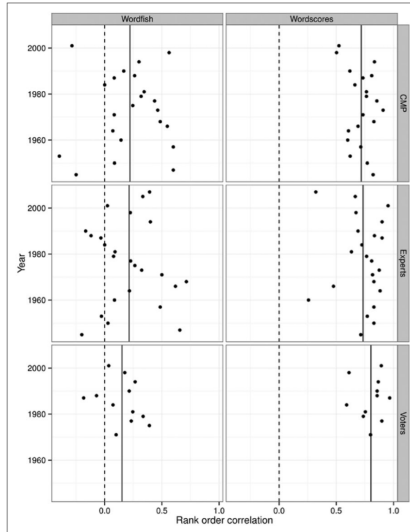


Figure 1. Wordfish and Wordscores estimates' rank order correlations with CMP, expert and voter estimates for each election year in the Danish sample. Vertical lines signify average rank order correlation across years.

German results

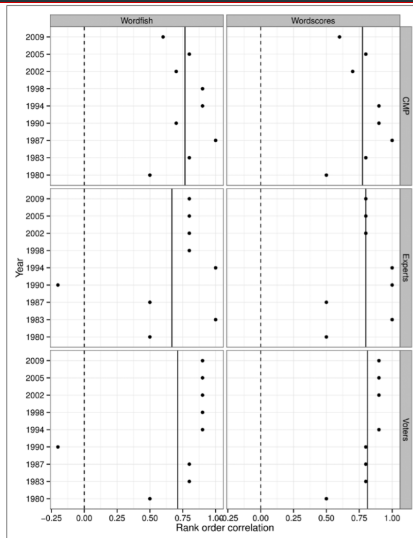


Figure 2. Wordfish and Wordscores estimates' rank order correlations with CMP, expert and voter estimates for each election year in the German sample. Vertical lines signify average rank order correlation across years.

- If you have **strong priors** about the location of certain actors, then use Wordscores
- If you have **no strong priors**
 1. Long and ideologically polarized texts: use Wordfish
 2. Short and ideologically similar texts: gather more data

Some of these recommendations may be outdated with newer **semi-supervised methods** of estimating Word positions

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

Latent Semantic Scaling

Semi-supervised method of scaling developed by Watanabe (2020). Procedure is as follows:

1. Segment a corpus at the sentence level (in effect, create a bag of words with each document a sentence)
2. Identify seed words
3. Estimate **semantic proximity of words by employing word-embedding techniques** (Singular Value Decomposition) on this corpus
4. Calculate **polarity scores** of words based on semantic proximity with seed words
5. Calculate polarity of documents by aggregating polarity scores for each document

```
> seed <- as.seedwords(data_dictionary_sentiment)
```

```
> print(seed)
```

good	nice	excellent	positive	fortunate	correct	superior		
1	1	1		1	1		1	1
bad	nasty	poor	negative	unfortunate	wrong	inferior		
-1	-1		-1	-1	-1		-1	-1

Latent Semantic Scaling

LSS resembles Wordscores in that it locates documents on a unidimensional scale by producing polarity scores of words, but these scores are computed based on their semantic proximity to seed words instead of their frequency in manually-coded documents; it automatically estimates semantic proximity between words in a corpus employing word-embedding techniques but users still have to choose seed words manually based on their knowledge. (Watanabe, 2020)

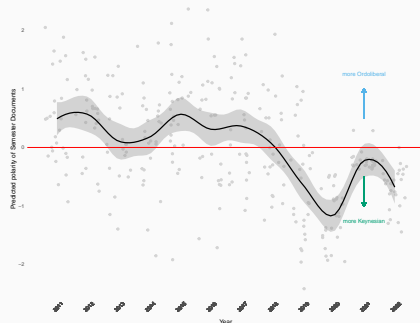
LSS is semi-supervised: you can rely on your **domain knowledge** to identify a small set of seed words to identify a relevant dimension

NB: requires a large corpus to estimate **word embeddings** for calculating polarity scores. Pre-trained word embeddings may not be useful for a domain-specific corpus

Latent Semantic Scaling: application

Graham *et al.* (2023) study the extent to which economic policy documents produced by the European Commission reflect more **Keynesian** or **Ordoliberal** ideas

- Corpus of 317 **country-specific recommendations** on economic policy that the Commission writes for EU member states
- Seed words: 80 Keynesian and Ordoliberal keywords, validated by domain experts



Source: Graham *et al.* 2023