university of groningen

Quantitative Text Analysis – Essex Summer School

Scaling methods

dr. Martijn Schoonvelde

University of Groningen

Today's class

- Learn how to use text data to estimate document positions on an underlying (latent) dimension
- **Scaling methods** (unidimensional):
  - **Wordscores** (Supervised)
  - **Wordfish** (Unsupervised)
  - **Latent Semantic Scaling** (Semi-supervised)
- Flash talk: Yeon
- Lab session

Motivation for scaling methods

- Research task: place documents or actors on a single ideological dimension
- Traditionally measured using:
  - Expert surveys (e.g. CHES)
  - Hand-coded content analysis (e.g. CMP)
  - Roll-call votes (parliaments)
- Limitations: expensive, labor-intensive, not always available
- **Idea: use the text that actors produce to estimate positions automatically**

Three broad approaches

Method	Type	Input Needed
Wordscores	Supervised	Reference texts + scores
Wordfish	Unsupervised	Just a DFM
LSS	Semi-supervised	Seed words + word embeddings

Assumption behind text scaling methods

- Ideological dominance assumption (Grimmer & Stewart 2013)
- Word use varies systematically with ideological position
- E.g., **"death tax" vs. "estate tax"** in U.S. politics

  Republicans... had to train members to say "death tax"... Estate tax sounds like it only hits the wealthy but "death tax" sounds like it hits everyone. (Graetz & Shapiro 2005)

- Not all texts follow this logic $\rightarrow$ validate carefully!

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 (numeric scores) to these texts (e.g., -1,1)
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						Probability of Reading Text r_i Given Reading Word w					Score	Virgin Score	
	Reference Text					Virgin								
Word w	r_1	r_2	r_3	r_4	r_5	Text	P_{w1}	P_{w2}	P_{w3}	P_{w4}	P_{w5}	S_{wr}	F_{wv}	$F_{wv} * S_{wr}$
A	2	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	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
C	19	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	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
E	45	0	0	0	0	0	1.00	0.00	0.00	0.00	0.00	−1.50	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
G	115	3	0	0	0	0	0.97	0.03	0.00	0.00	0.00	−1.48	0.0000	0.0000
H	146	10	0	0	0	2	0.94	0.06	0.00	0.00	0.00	−1.45	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	3	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	0.58	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_{wr}	−0.45
Estimated weighted variance V_{wr}	0.14
Estimated SD $\sqrt{V_{wr}}$	0.38
Estimated SE $\sqrt{\sum_w V_{wr}}$	0.018

Source: Laver, Benoit & Garry, 2003

Criticisms of Wordscores

- Relies heavily on the choice of reference texts
- Can conflate style or genre with ideology
- Lacks a generative model for text (Lowe 2008)
- Reference scores may not transfer across time
- See Bruinsma & Gemenis (2019) for detailed 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
```

Wordfish

- 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.

Application

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

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

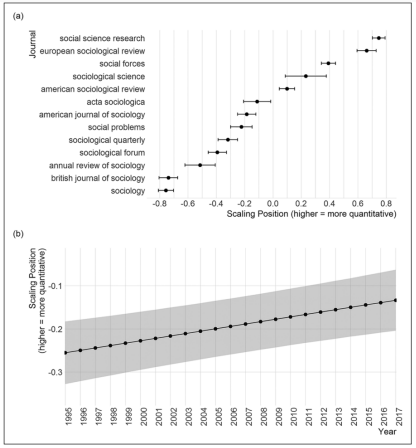


Figure 3. Predicted values for abstract scaling positions by journal (a) and publication year (b) with 95% confidence intervals.

Source: Schwemmer & Wieczorek, 2020

Wordfish Model: Application

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

Applied to the Sociology abstracts, this models the number of times abstract i inludes 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**.

How Wordfish is estimated

The Wordfish model is estimated using an Expectation-Maximization (EM) algorithm, a common method for fitting models with latent variables.

Steps in the EM algorithm:

1. **Initialize parameters:**
   Start with reasonable guesses for document positions (θ), word effects (ψ), etc.
2. **Estimate document parameters (θ and α):**
   Keep word parameters fixed and update document positions and fixed effects.
3. **Estimate word parameters (β and ψ):**
   Now keep the updated document parameters fixed and update word weights and fixed effects.
4. **Repeat steps 2 and 3 until convergence:**
   The process iterates until parameter estimates stabilize (i.e., likelihood stops improving).

Goal: Find document and word positions that best explain observed word counts.

Wordfish model: documents

Schwemmer & Wieczorek (2020) apply Wordfish to 8,700 sociology article abstracts (1995 – 2017).

- Estimated θ scores represent each document's position on a latent **quantitative–qualitative** dimension.
- Higher $\theta \rightarrow$ more quantitative.

1. Is there a systematic divide between qualitative and quantitative work?
2. Is the discipline becoming more quantitative over time?

They answer these by modeling θ across journals and years.

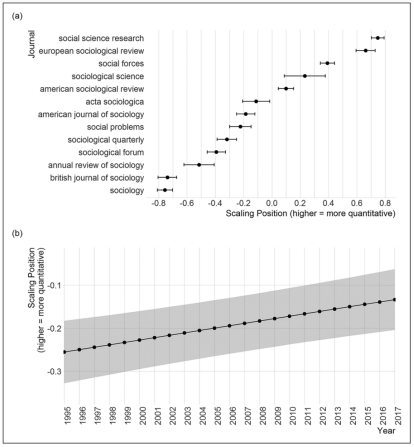


Figure 3. Predicted values for abstract scaling positions by journal (a) and publication year (b) with 95% confidence intervals.

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)

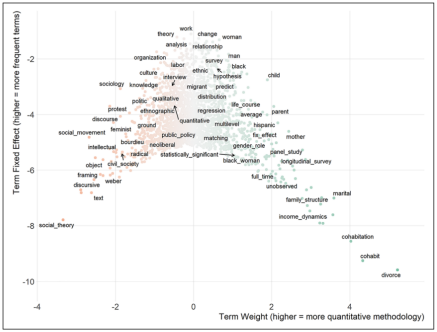


Figure 2. Term weights and fixed effects from the wordfish scaling model fitted to Sociology abstracts.

Source: Schwemmer & Wieczorek, 2020

Validating Wordfish Estimates

Wordfish provides positions – **but are they substantively meaningful?**

Validation is crucial because the estimated dimension is not labeled or guaranteed to be what you expect it to be.

Checklist for validating results:

- **Inspect term plots (e.g. Eiffel plot):**
  What are the most discriminating words? Do they reflect an ideological or thematic divide?
- **Read documents at both extremes (θ high and low):**
  Does the language used support your interpretation of the scale?
- **Compare to external benchmarks:**
  Use human-coded labels, expert surveys, or metadata (e.g., party affiliation, journal type).
- **Consider preprocessing effects (Denny & Spirling, 2018):**
  How sensitive are results to choices like stopword removal or stemming?

Remember: the model will always return a dimension – even when none exists.

Validating Scaling Models: Hjorth et al. (2015)

Assess how well Wordscores and Wordfish estimate party positions from manifestos.

- Applied to German and Danish party manifestos (1990s – 2010s)
- Compared model estimates to multiple external benchmarks:
  - **CHES expert surveys**
  - **Eurobarometer voter placements:** average self-placement of voters by party support
  - **CMP RILE index:** hand-coded ideological content
- Explored the effects of:
  - Number of parties per election
  - Length and richness of manifestos
  - Cross-national variation

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

Source: Hjorth et al. (2015)

Danish Manifestos: Validation Results

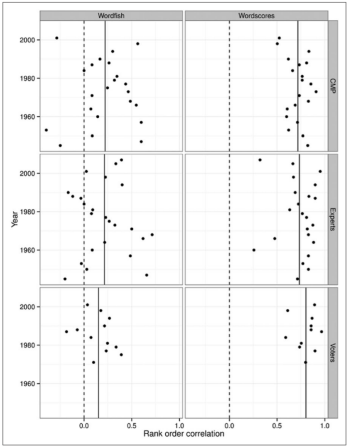


Figure 1. Wordfish and Wordscores estimates' rank order correlations with CMP, expert and voter estimates for each election year in the Danish sample. Vertical lines signify average rank order correlation across years.

Source: Hjorth et al. (2015)

German results

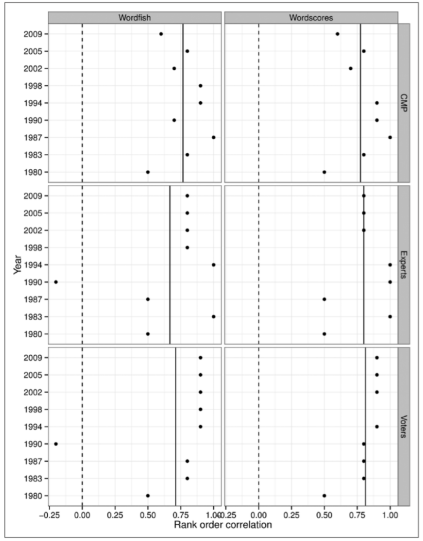


Figure 2. Wordfish and Wordscores estimates' rank order correlations with CMP, expert and voter estimates for each election year in the German sample. Vertical lines signify average rank order correlation across years.

Recommendations Hjorth et al. (2015)

Guidelines based on text characteristics and available priors:

- Use Wordscores if:
  - You have **strong prior knowledge** about the ideological positions of reference texts
  - Your documents are relatively short or constrained (e.g., manifestos)
- Use Wordfish if:
  - You have **no strong priors**
  - Your documents are **long** and **ideologically polarized**
- If documents are **short** and **ideologically similar**:
  - Neither method may work well – consider gathering more data or switching methods

Note: These recommendations may need updating in light of newer semi-supervised methods (e.g., LSS), which combine domain knowledge with data-driven scaling.

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

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 on this corpus
4. Calculate polarity scores of context 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

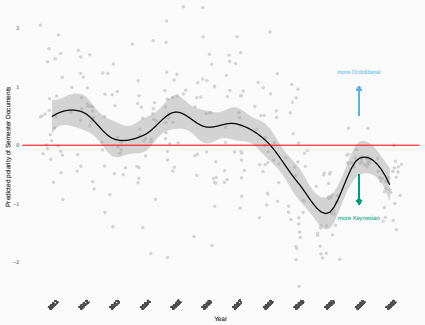
LSS in Practice: Economic Ideas in EU Policy

Do EU policy documents promote more Keynesian or Ordoliberal ideas?

Graham *et al.* (2024) analyze a corpus of 317 **country-specific economic policy recommendations** issued by the European Commission.

Method:

- Seed words: 80 expert-validated terms linked to **Keynesian** (e.g., stimulus, investment) and **Ordoliberal** (e.g., austerity, rules) thinking
- LSS computes polarity scores for each document based on semantic similarity to the seeds



Source: Graham *et al.* (2023)