Supervised and Unsupervised Scaling Models: Wordscores and Wordfish

2023-10-11

```
myText <-
readtext("C:/Users/Miras/Desktop/u_m/1st/big_data_analytics/Labs/projects/uk_
new_manifestos/*.txt",
                   docvarsfrom = "filenames", dvsep = " ", docvarnames =
c("Party", "Year"))
glimpse(myText)
```

```
## Rows: 12
## Columns: 4
## $ doc_id <chr> "Cons 2010.txt", "Cons 2015.txt", "Cons 2017.txt", "Lab
2010.tx…
## $ text   <chr> "THE CONSERVATIVE MANIFESTO 2010\nINVITATION TO\nJOIN THE
GOVER…
## $ Party  <chr> "Cons", "Cons", "Cons", "Lab", "Lab", "Lab", "Lib", "Lib",
"Lib…
## $ Year   <int> 2010, 2015, 2017, 2010, 2015, 2017, 2010, 2015, 2017, 2010,
201…
```

```
myText$text <- gsub("'"," ",myText$text)
myText$text <- gsub("[\u00E2]"," ",myText$text)

testCorpus <- corpus(myText)
summary(testCorpus)
```

```
## Corpus consisting of 12 documents, showing 12 documents:
##
##           Text Types Tokens Sentences Party Year
##  Cons 2010.txt  4316  30568      1116  Cons 2010
##  Cons 2015.txt  4429  32976      1110  Cons 2015
##  Cons 2017.txt  4177  32657      1184  Cons 2017
##   Lab 2010.txt  4389  33046      1301   Lab 2010
##   Lab 2015.txt  3157  19818       859   Lab 2015
##   Lab 2017.txt  4291  25797      1033   Lab 2017
##   Lib 2010.txt  3424  20541       776   Lib 2010
##   Lib 2015.txt  5034  37555       759   Lib 2015
##   Lib 2017.txt  4081  23464       464   Lib 2017
##  UKIP 2010.txt  2465   9103       236  UKIP 2010
##  UKIP 2015.txt  5470  30044      1059  UKIP 2015
##  UKIP 2017.txt  5457  27581       999  UKIP 2017
```

```
testCorpus <- corpus_subset(testCorpus, Year > 2010)
summary(testCorpus)
```

```
## Corpus consisting of 8 documents, showing 8 documents:
##
##           Text Types Tokens Sentences Party Year
##  Cons 2015.txt  4429  32976      1110  Cons 2015
##  Cons 2017.txt  4177  32657      1184  Cons 2017
##   Lab 2015.txt  3157  19818       859   Lab 2015
##   Lab 2017.txt  4291  25797      1033   Lab 2017
##   Lib 2015.txt  5034  37555       759   Lib 2015
##   Lib 2017.txt  4081  23464       464   Lib 2017
##  UKIP 2015.txt  5470  30044      1059  UKIP 2015
##  UKIP 2017.txt  5457  27581       999  UKIP 2017

tok1 <- tokens(testCorpus, remove_punct = TRUE, remove_numbers=TRUE,
remove_symbols = TRUE, split_hyphens = TRUE, remove_separators = TRUE)
tok1 <- tokens_remove(tok1, stopwords("en"))
tok1 <- tokens_wordstem (tok1)
myDfm <- dfm(tok1)
topfeatures(myDfm , 20)

##   peopl    work     new support  govern    year  servic   ensur    need
nation
##     994     982     755     744     692     640     626     617     603
597
##    make  public    help   local     can   right    care  labour      eu
tax
##     586     527     525     524     513     504     496     491     481
472

myDfm <- dfm_remove(myDfm, min_nchar=2)
topfeatures(myDfm , 20)

##   peopl    work     new support  govern    year  servic   ensur    need
nation
##     994     982     755     744     692     640     626     617     603
597
##    make  public    help   local     can   right    care  labour      eu
tax
##     586     527     525     524     513     504     496     491     481
472

Simil <- textstat_simil(myDfm , method = "cosine")
Simil

## textstat_simil object; method = "cosine"
##               Cons 2015.txt Cons 2017.txt Lab 2015.txt Lab 2017.txt
## Cons 2015.txt         1.000         0.876        0.864        0.753
## Cons 2017.txt         0.876         1.000        0.866        0.772
## Lab 2015.txt          0.864         0.866        1.000        0.837
## Lab 2017.txt          0.753         0.772        0.837        1.000
## Lib 2015.txt          0.858         0.869        0.852        0.769
## Lib 2017.txt          0.823         0.858        0.835        0.774
```

```
## UKIP 2015.txt         0.754         0.752        0.723        0.660
## UKIP 2017.txt         0.741         0.744        0.718        0.656
##               Lib 2015.txt Lib 2017.txt UKIP 2015.txt UKIP 2017.txt
## Cons 2015.txt        0.858        0.823         0.754         0.741
## Cons 2017.txt        0.869        0.858         0.752         0.744
## Lab 2015.txt         0.852        0.835         0.723         0.718
## Lab 2017.txt         0.769        0.774         0.660         0.656
## Lib 2015.txt         1.000        0.938         0.737         0.721
## Lib 2017.txt         0.938        1.000         0.745         0.728
## UKIP 2015.txt        0.737        0.745         1.000         0.930
## UKIP 2017.txt        0.721        0.728         0.930         1.000
```

```
Simil[c(1,3,5),c(1,3,5)] #similarity > 0.6
```

```
## 3 x 3 Matrix of class "dspMatrix"
##               Cons 2015.txt Lab 2015.txt Lib 2015.txt
## Cons 2015.txt     1.0000000    0.8639989    0.8584731
## Lab 2015.txt      0.8639989    1.0000000    0.8516368
## Lib 2015.txt      0.8584731    0.8516368    1.0000000
```

1. Wordscore supervised scaling model

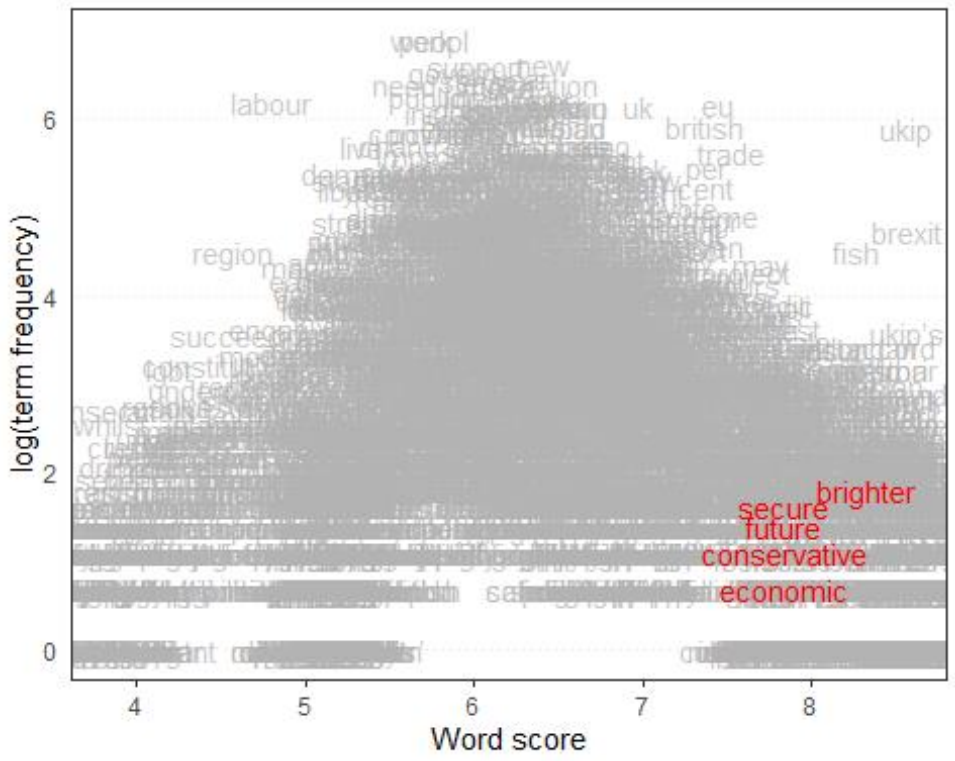
```
# reference texts are 2015 parties manifestos: Economic dimension (scores
refer to a left-right economic scale):
#CONS=7.85; Lab=3.85; Lib=5.14; UKIP=8.57 [source of parties' scores: 2014
Chapel Hill expert survey]
docnames(myDfm)
```

```
## [1] "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt"  "Lab 2017.txt"
## [5] "Lib 2015.txt"  "Lib 2017.txt"  "UKIP 2015.txt" "UKIP 2017.txt"
```

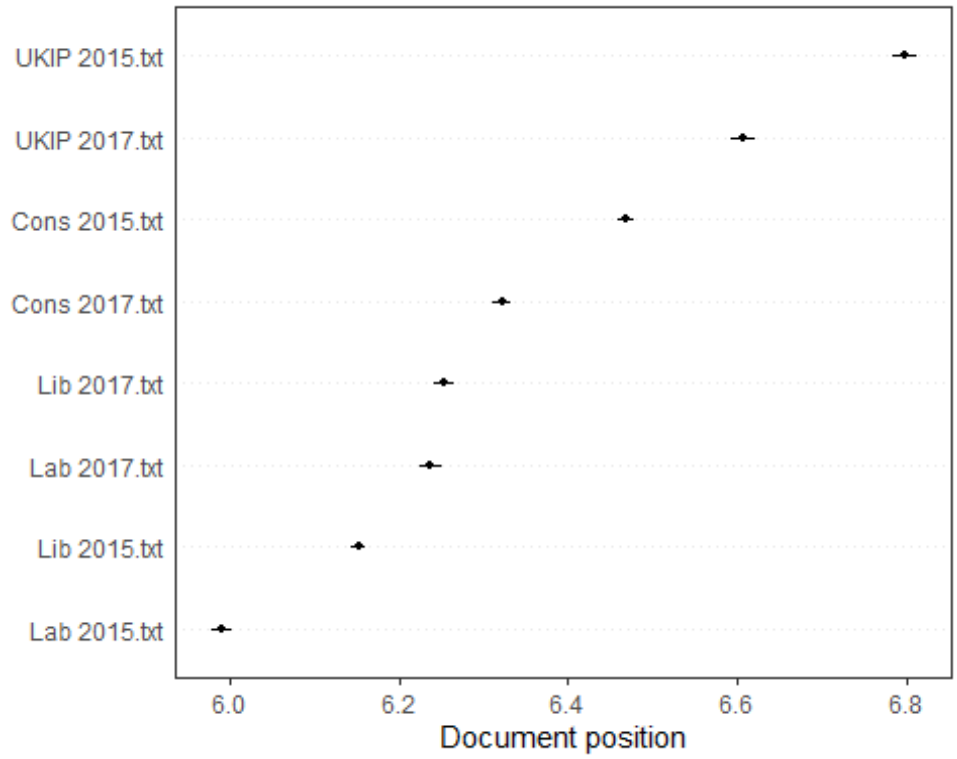
```
# reference texts are 1st, 3rd, 5th, 7th.
ws <- textmodel_wordscores(myDfm, c(7.85, NA, 3.85, NA, 5.14, NA, 8.57, NA))
summary(ws)
```

```
##
## Call:
## textmodel_wordscores.dfm(x = myDfm, y = c(7.85, NA, 3.85, NA,
##     5.14, NA, 8.57, NA))
##
## Reference Document Statistics:
##               score total min max  mean median
## Cons 2015.txt  7.85 16513   0 176 2.658      0
## Cons 2017.txt    NA 16459   0 178 2.649      0
## Lab 2015.txt   3.85 10082   0 147 1.623      0
## Lab 2017.txt     NA 13778   0 319 2.218      0
## Lib 2015.txt   5.14 20158   0 209 3.244      0
## Lib 2017.txt     NA 12675   0 113 2.040      0
## UKIP 2015.txt  8.57 15469   0 166 2.490      0
```

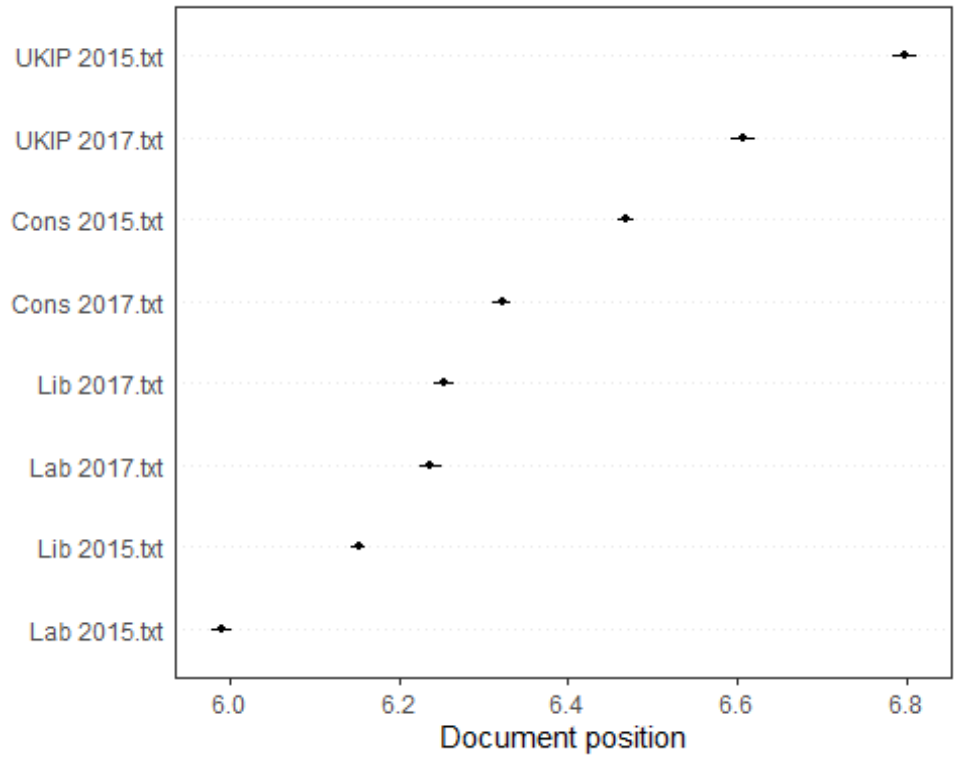
```
## UKIP 2017.txt    NA 14544   0 199 2.341      1
##
## Wordscores:
## (showing first 30 elements)
## conservative        parti    manifesto        stong   leadership
clear
##        7.850        6.598        5.120        7.850        5.669
6.559
##     economic         plan     brighter       secure       future
everi
##        7.850        6.624        8.340        7.850        7.850
6.060
##        stage         life         best        start      continu
increas
##        6.994        5.859        6.426        6.119        6.512
6.258
##        spend          nhs       provid          day         week
access
##        6.689        6.273        6.195        7.096        6.036
6.148
##           gp        deliv        truli         know        alway
free
##        6.333        6.236        7.568        5.550        6.133
6.794

textplot_scale1d(ws,  margin = "features",
                 highlighted = c( "conservative", "secure", "economic",
"brighter", "future" ),
                 highlighted_color = "red")   #the words with highest scores
were highlighted. Do they represent right political views? Not certainly
because only 8 documents are being analyzed.
```



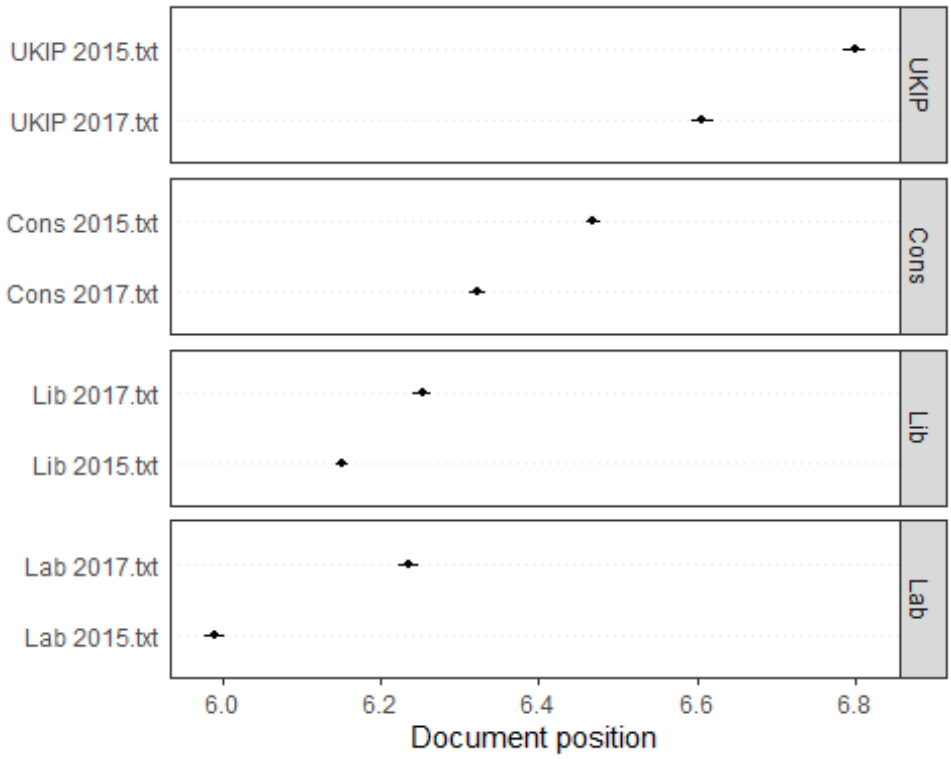

```
pr_raw <- predict(ws, se.fit = TRUE, newdata = myDfm)

## Warning: 1456 features in newdata not used in prediction.

pr_raw

## $fit
## Cons 2015.txt Cons 2017.txt  Lab 2015.txt  Lab 2017.txt  Lib 2015.txt
##      6.468983      6.322198      5.990080      6.236917      6.152116
##  Lib 2017.txt UKIP 2015.txt UKIP 2017.txt
##      6.252537      6.798822      6.606603
##
## $se.fit
## [1] 0.005036316 0.005350599 0.006415195 0.006468352 0.004159776
0.005974366
## [7] 0.006950157 0.007340189

textplot_scale1d(pr_raw)
```




```
# alternative way (with c.i. rather than with s.e.)
pr_all <- predict(ws, interval = "confidence", newdata = myDfm)

## Warning: 1456 features in newdata not used in prediction.

pr_all

## $fit
##                      fit      lwr      upr
## Cons 2015.txt 6.468983 6.459112 6.478854
## Cons 2017.txt 6.322198 6.311711 6.332685
## Lab 2015.txt  5.990080 5.977506 6.002653
## Lab 2017.txt  6.236917 6.224239 6.249594
## Lib 2015.txt  6.152116 6.143963 6.160269
## Lib 2017.txt  6.252537 6.240827 6.264246
## UKIP 2015.txt 6.798822 6.785200 6.812444
## UKIP 2017.txt 6.606603 6.592217 6.620990

textplot_scale1d(pr_all)
```




```
# Plot estimated document positions and group by "party" variable
summary(testCorpus)

## Corpus consisting of 8 documents, showing 8 documents:
##
##           Text Types Tokens Sentences Party Year
##  Cons 2015.txt  4429  32976      1110  Cons 2015
##  Cons 2017.txt  4177  32657      1184  Cons 2017
##   Lab 2015.txt  3157  19818       859   Lab 2015
##   Lab 2017.txt  4291  25797      1033   Lab 2017
##   Lib 2015.txt  5034  37555       759   Lib 2015
##   Lib 2017.txt  4081  23464       464   Lib 2017
##  UKIP 2015.txt  5470  30044      1059  UKIP 2015
##  UKIP 2017.txt  5457  27581       999  UKIP 2017

textplot_scale1d(pr_all,  groups = docvars(testCorpus, "Party"))
```




```
# we want to predict only the virgin texts using the rescaling LGB option. Because raw scores are dispersed on a much smaller scale, they cannot therefore be directly compared to the exogenous scores attached to the reference texts
summary(ws)

##
## Call:
## textmodel_wordscores.dfm(x = myDfm, y = c(7.85, NA, 3.85, NA,
##     5.14, NA, 8.57, NA))
##
## Reference Document Statistics:
##               score total min max  mean median
## Cons 2015.txt  7.85 16513   0 176 2.658      0
## Cons 2017.txt    NA 16459   0 178 2.649      0
## Lab 2015.txt   3.85 10082   0 147 1.623      0
## Lab 2017.txt     NA 13778   0 319 2.218      0
## Lib 2015.txt   5.14 20158   0 209 3.244      0
## Lib 2017.txt     NA 12675   0 113 2.040      0
## UKIP 2015.txt  8.57 15469   0 166 2.490      0
## UKIP 2017.txt    NA 14544   0 199 2.341      1
##
## Wordscores:
## (showing first 30 elements)
## conservative        parti    manifesto        stong   leadership
clear
##        7.850        6.598        5.120        7.850        5.669
```

```
6.559
##     economic          plan     brighter       secure       future
everi
##        7.850         6.624        8.340        7.850        7.850
6.060
##        stage          life         best        start      continu
increas
##        6.994         5.859        6.426        6.119        6.512
6.258
##        spend           nhs       provid          day         week
access
##        6.689         6.273        6.195        7.096        6.036
6.148
##           gp         deliv        truli         know        alway
free
##        6.333         6.236        7.568        5.550        6.133
6.794
```

```
pr_lbg <- predict(ws, rescaling = "lbg", newdata = myDfm[c(2, 4, 6, 8), ])
```

```
## Warning: 1456 features in newdata not used in prediction.
```

```
pr_lbg
```

```
## Cons 2017.txt  Lab 2017.txt  Lib 2017.txt UKIP 2017.txt
##      5.935473      4.831196      5.033456      9.618130
```

Apparently, UKIP is positioned on the extreme right position on the left-right spectrum related to Economic policies, while Labour party is the leftest one not very much differing from Conservatives and Liberals in 2017.

```
# reference texts are 2015 parties manifestos: EU dimension(higher score
implies being more pro-EU):
#CONS=3.14; Lab=5.57; Lib=6.71; UKIP=1.14 [source of parties' scores: 2014
Chapel Hill expert survey]
docnames(myDfm)
```

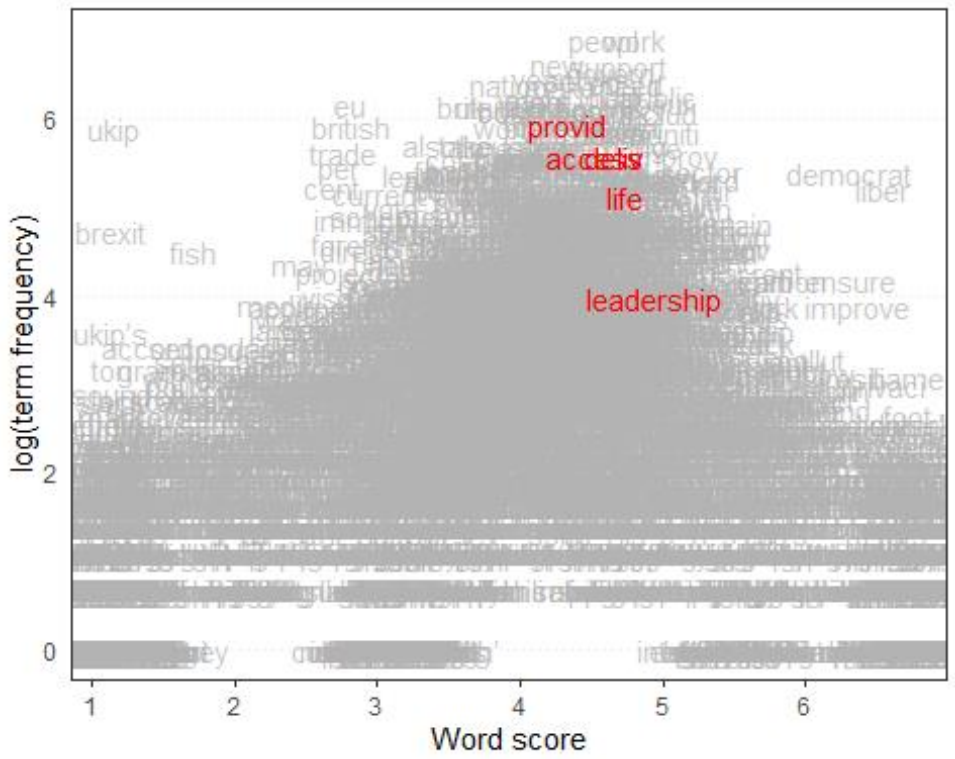
```
## [1] "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt"  "Lab 2017.txt"
## [5] "Lib 2015.txt"  "Lib 2017.txt"  "UKIP 2015.txt" "UKIP 2017.txt"
```

```
# reference texts are 1st, 3rd, 5th, 7th.
ws2 <- textmodel_wordscores(myDfm, c(3.14, NA, 5.57, NA, 6.71, NA, 1.14, NA))
summary(ws2)
```

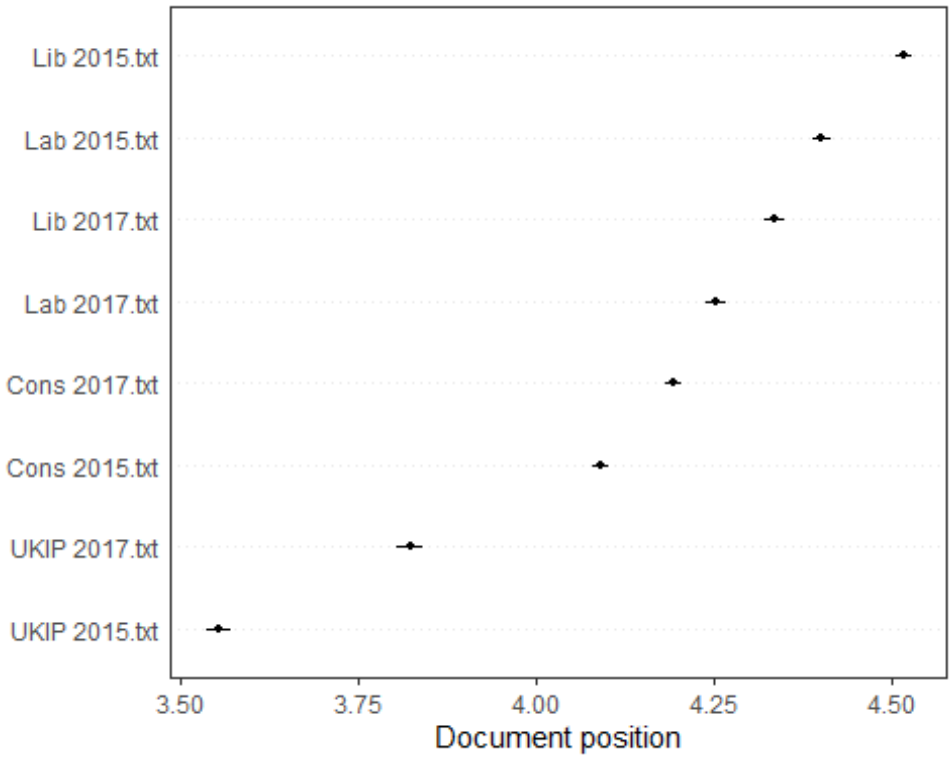
```
##
## Call:
## textmodel_wordscores.dfm(x = myDfm, y = c(3.14, NA, 5.57, NA,
##     6.71, NA, 1.14, NA))
##
## Reference Document Statistics:
##                score total min max  mean median
## Cons 2015.txt  3.14 16513   0 176 2.658      0
```

```
## Cons 2017.txt    NA 16459   0 178 2.649      0
## Lab 2015.txt   5.57 10082   0 147 1.623      0
## Lab 2017.txt     NA 13778   0 319 2.218      0
## Lib 2015.txt   6.71 20158   0 209 3.244      0
## Lib 2017.txt     NA 12675   0 113 2.040      0
## UKIP 2015.txt  1.14 15469   0 166 2.490      0
## UKIP 2017.txt    NA 14544   0 199 2.341      1
##
## Wordscores:
## (showing first 30 elements)
## conservative        parti    manifesto        stong   leadership
clear
##        3.140        3.770        4.785        3.140        4.944
4.010
##     economic         plan     brighter       secure       future
everi
##        3.140        4.089        1.778        3.140        3.140
4.332
##        stage         life         best        start      continu
increas
##        3.820        4.733        4.032        4.407        4.264
4.319
##        spend          nhs       provid          day         week
access
##        3.845        4.154        4.344        3.388        4.159
4.526
##           gp        deliv        truli         know        alway
free
##        4.044        4.663        3.158        4.522        4.511
3.853
```

```
textplot_scale1d(ws2,  margin = "features",
                 highlighted = c( "leadership", "deliv", "life", "access",
"provid" ),
                 highlighted_color = "red") #example of highest scoring words
were highlighted. Do they represent pro-EU policy perspective? Difficult to
say because only  documents are being analyzed.
```




```
pr_raw2 <- predict(ws2, se.fit = TRUE, newdata = myDfm)

## Warning: 1456 features in newdata not used in prediction.

pr_raw2

## $fit
## Cons 2015.txt Cons 2017.txt  Lab 2015.txt  Lab 2017.txt  Lib 2015.txt
##      4.090247      4.192612      4.401411      4.251584      4.516285
##  Lib 2017.txt UKIP 2015.txt UKIP 2017.txt
##      4.334150      3.552057      3.821773
##
## $se.fit
## [1] 0.005169864 0.006165627 0.005948219 0.007149658 0.005627647
0.007589120
## [7] 0.008659211 0.009042926

textplot_scale1d(pr_raw2)
```




```
summary(ws)

##
## Call:
## textmodel_wordscores.dfm(x = myDfm, y = c(7.85, NA, 3.85, NA,
##     5.14, NA, 8.57, NA))
##
## Reference Document Statistics:
##               score total min max  mean median
## Cons 2015.txt  7.85 16513   0 176 2.658      0
## Cons 2017.txt    NA 16459   0 178 2.649      0
## Lab 2015.txt   3.85 10082   0 147 1.623      0
## Lab 2017.txt     NA 13778   0 319 2.218      0
## Lib 2015.txt   5.14 20158   0 209 3.244      0
## Lib 2017.txt     NA 12675   0 113 2.040      0
## UKIP 2015.txt  8.57 15469   0 166 2.490      0
## UKIP 2017.txt    NA 14544   0 199 2.341      1
##
## Wordscores:
## (showing first 30 elements)
## conservative        parti    manifesto        stong   leadership
clear
##        7.850        6.598        5.120        7.850        5.669
6.559
##     economic         plan     brighter       secure       future
everi
##        7.850        6.624        8.340        7.850        7.850
```

```
6.060
##        stage         life         best        start      continu
increas
##        6.994        5.859        6.426        6.119        6.512
6.258
##        spend          nhs       provid          day         week
access
##        6.689        6.273        6.195        7.096        6.036
6.148
##           gp        deliv        truli         know        alway
free
##        6.333        6.236        7.568        5.550        6.133
6.794
```

```
pr_lbg2 <- predict(ws2, rescaling = "lbg", newdata = myDfm[c(2, 4, 6, 8), ])
```

```
## Warning: 1456 features in newdata not used in prediction.
```

```
pr_lbg2
```

```
## Cons 2017.txt  Lab 2017.txt  Lib 2017.txt UKIP 2017.txt
##      4.618968      5.268400      6.177667      0.535084
```

Apparently, UKIP is least pro-EU, while Liberals are the biggest pro-EU party in 2017.

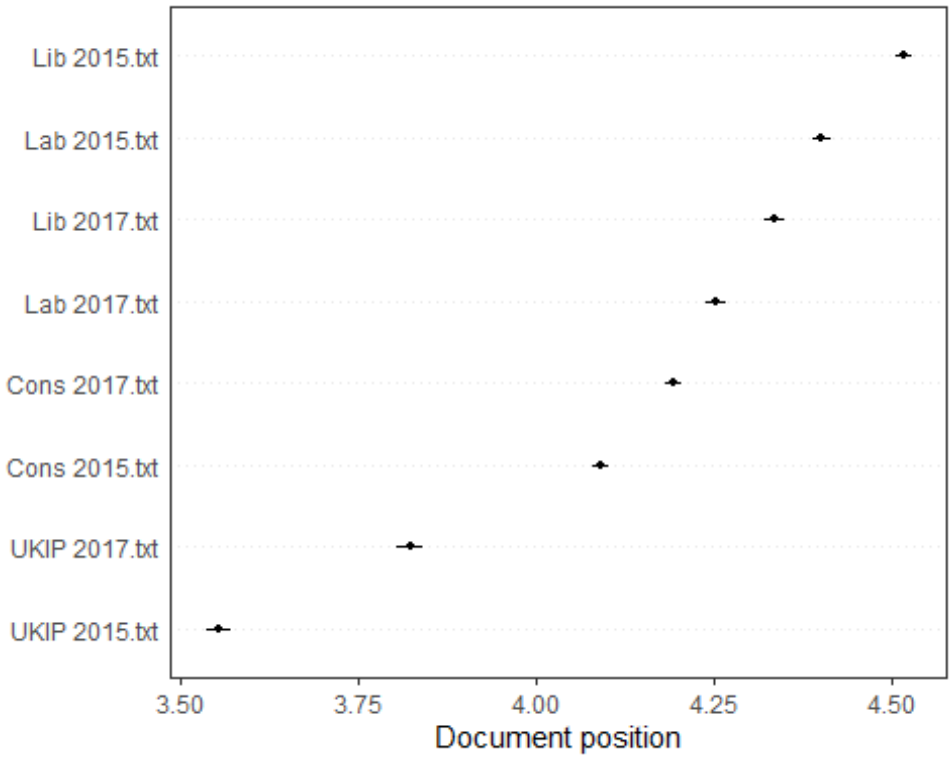
```
# alternative way (with c.i. rather than with s.e.)
pr_all2 <- predict(ws2, interval = "confidence", newdata = myDfm)
```

```
## Warning: 1456 features in newdata not used in prediction.
```

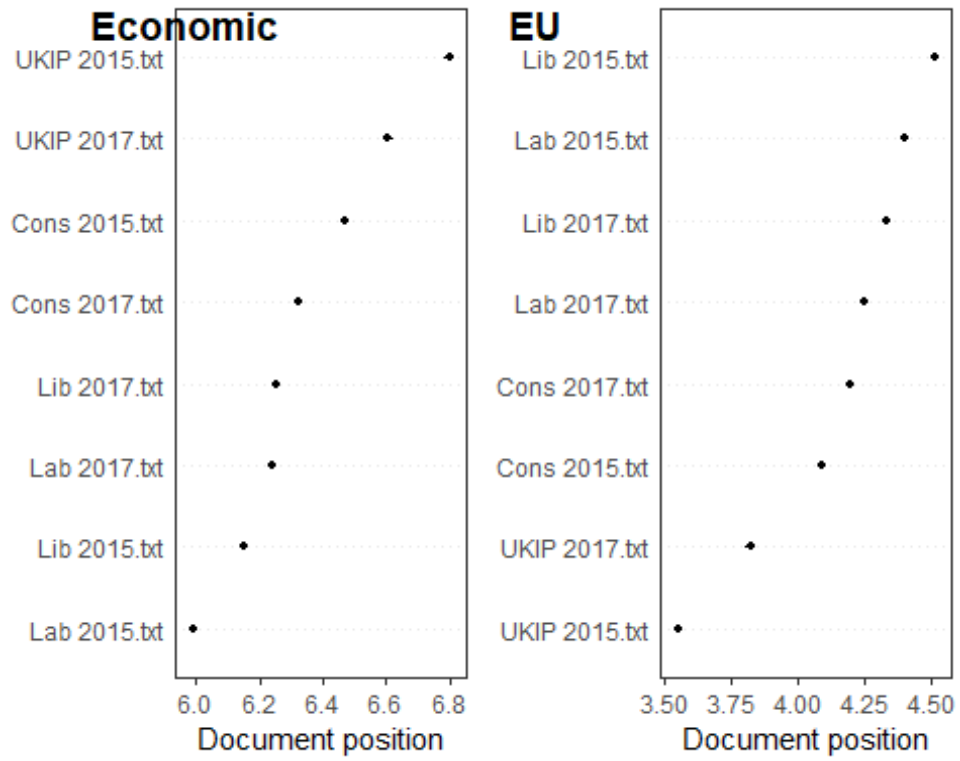
```
pr_all2
```

```
## $fit
##                      fit      lwr      upr
## Cons 2015.txt 4.090247 4.080114 4.100379
## Cons 2017.txt 4.192612 4.180527 4.204696
## Lab 2015.txt  4.401411 4.389753 4.413069
## Lab 2017.txt  4.251584 4.237571 4.265597
## Lib 2015.txt  4.516285 4.505255 4.527315
## Lib 2017.txt  4.334150 4.319276 4.349024
## UKIP 2015.txt 3.552057 3.535086 3.569029
## UKIP 2017.txt 3.821773 3.804050 3.839497
```

```
textplot_scale1d(pr_all2)
```




```
pr_lbg2 <- predict(ws2, rescaling = "lbg", newdata = myDfm[c(2, 4, 6, 8), ],
interval = "confidence")

## Warning: 1456 features in newdata not used in prediction.

pr_lbg2

## $fit
##                     fit       lwr       upr
## Cons 2017.txt 4.618968 4.4858876 4.7520483
## Lab 2017.txt  5.268400 5.1140805 5.4227202
## Lib 2017.txt  6.177667 6.0138614 6.3414721
## UKIP 2017.txt 0.535084 0.3398994 0.7302687

#comparison: economic VS EU dimensions
eco <- textplot_scale1d(pr_all)
EU <- textplot_scale1d(pr_all2)

plot_grid(eco , EU , labels = c('Economic', 'EU'))
```

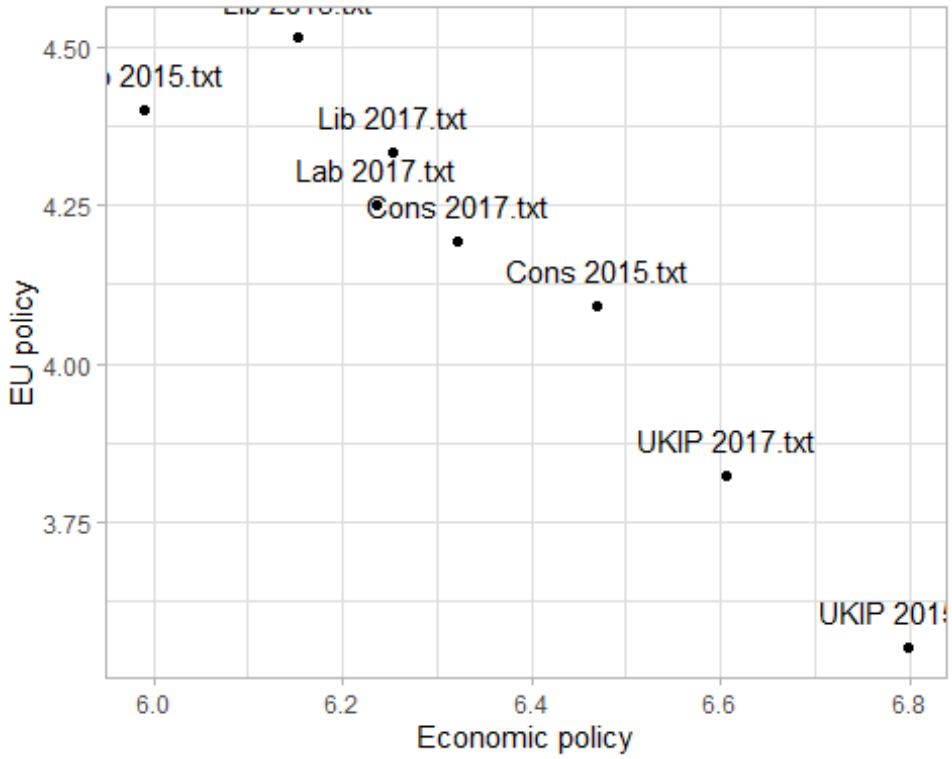



```
str(ws)
```

```
## List of 5
##  $ wordscores: Named num [1:4757] 7.85 6.6 5.12 7.85 5.67 ...
##   ..- attr(*, "names")= chr [1:4757] "conservative" "parti" "manifesto"
"stong" ...
##  $ x         :Formal class 'dfm' [package "quanteda"] with 8 slots
##   .. ..@ docvars :'data.frame':  8 obs. of  5 variables:
##   .. .. ..$ docname_: chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab
2015.txt" "Lab 2017.txt" ...
##   .. .. ..$ docid_  : Factor w/ 8 levels "Cons 2015.txt",..: 1 2 3 4 5 6 7
8
##   .. .. ..$ segid_  : int [1:8] 1 1 1 1 1 1 1 1
##   .. .. ..$ Party   : chr [1:8] "Cons" "Cons" "Lab" "Lab" ...
##   .. .. ..$ Year    : int [1:8] 2015 2017 2015 2017 2015 2017 2015 2017
##   .. ..@ meta    :List of 3
##   .. .. ..$ system:List of 5
##   .. .. .. ..$ package-version:Classes 'package_version',
'numeric_version'  hidden list of 1
##   .. .. .. .. ..$ : int [1:3] 3 3 1
##   .. .. .. ..$ r-version      :Classes 'R_system_version',
'package_version', 'numeric_version'  hidden list of 1
##   .. .. .. .. ..$ : int [1:3] 4 3 1
##   .. .. .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "Miras"
##   .. .. .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
##   .. .. .. ..$ directory      : chr "C:/Users/Miras/Desktop/u Milan/1st
year classes/Big Data Analystics/Labs/Lab1"
```

```
##   .. .. .. ..$ created        : Date[1:1], format: "2023-10-11"
##   .. .. ..$ object:List of 9
##   .. .. .. ..$ unit        : chr "documents"
##   .. .. .. ..$ what        : chr "word"
##   .. .. .. ..$ ngram       : int 1
##   .. .. .. ..$ skip        : int 0
##   .. .. .. ..$ concatenator: chr "_"
##   .. .. .. ..$ weight_tf   :List of 3
##   .. .. .. .. ..$ scheme: chr "count"
##   .. .. .. .. ..$ base  : NULL
##   .. .. .. .. ..$ k     : NULL
##   .. .. .. ..$ weight_df   :List of 5
##   .. .. .. .. ..$ scheme   : chr "unary"
##   .. .. .. .. ..$ base     : NULL
##   .. .. .. .. ..$ c        : NULL
##   .. .. .. .. ..$ smoothing: NULL
##   .. .. .. .. ..$ threshold: NULL
##   .. .. .. ..$ smooth      : num 0
##   .. .. .. ..$ summary     :List of 2
##   .. .. .. .. ..$ hash: chr(0)
##   .. .. .. .. ..$ data: NULL
##   .. .. ..$ user  : list()
##   .. ..@ i       : int [1:20687] 0 1 0 1 2 3 4 5 6 7 ...
##   .. ..@ p       : int [1:6214] 0 2 10 18 19 27 35 36 44 47 ...
##   .. ..@ Dim     : int [1:2] 8 6213
##   .. ..@ Dimnames:List of 2
##   .. .. ..$ docs    : chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab
2015.txt" "Lab 2017.txt" ...
##   .. .. ..$ features: chr [1:6213] "conservative" "parti" "manifesto"
"stong" ...
##   .. ..@ x       : num [1:20687] 2 1 17 11 13 19 27 16 33 42 ...
##   .. ..@ factors : list()
##  $ y         : num [1:8] 7.85 NA 3.85 NA 5.14 NA 8.57 NA
##  $ scale     : chr "linear"
##  $ call      : language textmodel_wordscores.dfm(x = myDfm, y = c(7.85,
NA, 3.85, NA, 5.14, NA,      8.57, NA))
##  - attr(*, "class")= chr [1:3] "textmodel_wordscores" "textmodel" "list"

str(pr_all)

## List of 1
##  $ fit: num [1:8, 1:3] 6.47 6.32 5.99 6.24 6.15 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##   .. ..$ : chr [1:3] "fit" "lwr" "upr"
##  - attr(*, "class")= chr [1:2] "predict.textmodel_wordscores" "list"

str(pr_all2)
```

```
## List of 1
##  $ fit: num [1:8, 1:3] 4.09 4.19 4.4 4.25 4.52 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##   .. ..$ : chr [1:3] "fit" "lwr" "upr"
##  - attr(*, "class")= chr [1:2] "predict.textmodel_wordscores" "list"

# check for the correlation
party <- ws$x@Dimnames$docs
score_EU <- pr_all2$fit
score_eco <- pr_all$fit

scores_texts <-data.frame(party, score_EU, score_eco )
str(scores_texts)

## 'data.frame':    8 obs. of  7 variables:
##  $ party: chr  "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##  $ fit  : num  4.09 4.19 4.4 4.25 4.52 ...
##  $ lwr  : num  4.08 4.18 4.39 4.24 4.51 ...
##  $ upr  : num  4.1 4.2 4.41 4.27 4.53 ...
##  $ fit.1: num  6.47 6.32 5.99 6.24 6.15 ...
##  $ lwr.1: num  6.46 6.31 5.98 6.22 6.14 ...
##  $ upr.1: num  6.48 6.33 6 6.25 6.16 ...

colnames(scores_texts)[2] <- "scoreEU"
colnames(scores_texts)[5] <- "scoreECO"
str(scores_texts)

## 'data.frame':    8 obs. of  7 variables:
##  $ party   : chr  "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##  $ scoreEU : num  4.09 4.19 4.4 4.25 4.52 ...
##  $ lwr     : num  4.08 4.18 4.39 4.24 4.51 ...
##  $ upr     : num  4.1 4.2 4.41 4.27 4.53 ...
##  $ scoreECO: num  6.47 6.32 5.99 6.24 6.15 ...
##  $ lwr.1   : num  6.46 6.31 5.98 6.22 6.14 ...
##  $ upr.1   : num  6.48 6.33 6 6.25 6.16 ...

cor(scores_texts$scoreEU, scores_texts$scoreECO)

## [1] -0.9530348

# Plotting the 2-D policy space

ggplot(scores_texts, aes(x=scoreECO, y=scoreEU)) + geom_point() +
  geom_text(label=scores_texts$party, vjust=-1) +
  ylab(label="EU policy") +  xlab("Economic policy")  +
  theme_light()
```



As a result, we got scores for the virgin text (i.e. party manifestos for 2017). And in relation to the left-right economic dimension it is seen that UKIP and Conservatives shifted to the left in 2017 while Liberals and Labour parties shifted to the right along the scale. At the same time, while UKIP and Conservatives moved towards pro EU policy direction in 2017 in comparison to their positions in 2015, Liberals and Labour party shifted in opposite directions. Such positions of UK parties is not surprising given the political environment of 201 where EU was shocked with terrorist attacks in France, migrant influx, and most importantly for Brits promised Brexit referendum. As a result fo 2015 elections Conservatives won and took formed government.

2. Wordfish unsupervised scaling model

```
# here: lab 2015 to the left of UKIP 2015
wfm <- textmodel_wordfish(myDfm, dir = c(3, 7))
summary(wfm)

##
## Call:
## textmodel_wordfish.dfm(x = myDfm, dir = c(3, 7))
##
## Estimated Document Positions:
##                   theta      se
## Cons 2015.txt -0.08994 0.01991
## Cons 2017.txt -0.14865 0.01974
## Lab 2015.txt  -0.40978 0.02319
## Lab 2017.txt  -0.56030 0.01824
## Lib 2015.txt  -0.94550 0.01043
## Lib 2017.txt  -0.92188 0.01354
## UKIP 2015.txt  1.45712 0.01212
## UKIP 2017.txt  1.61893 0.01056
##
## Estimated Feature Scores:
##      conservative  parti manifesto    stong leadership    clear economic
plan
## beta      -0.1309 0.3341    0.4219 -0.09807    -0.4291 -0.04313  -0.1045 -
0.114
## psi       -1.0059 3.0324    2.1625 -2.10149     1.8033  2.46603  -1.4089
3.811
##      brighter  secure  future   everi    stage    life     best   start
continu
## beta    1.448  0.1680  0.5503 0.01878 -0.06819 -0.3822 -0.03243 -0.1376 -
0.2761
## psi    -1.296 -0.5009 -0.8560 3.57901  1.51146  2.9646  3.14376  2.8074
4.0102
##      increas  spend     nhs  provid    day     week  access     gp  deliv
## beta  -0.171 0.0304 0.08278 -0.1426 0.1472 -0.02294 -0.1943 0.1682 -0.417
## psi    3.841 3.3573 3.60645  3.8238 2.4878  2.17889  3.4708 1.3554  3.401
##         truli     know    alway    free
## beta 0.007392 -0.05351 -0.04499 0.05466
## psi  1.121273  1.92663  2.27107 3.46559

str(wfm)

## List of 13
##  $ x         :Formal class 'dfm' [package "quanteda"] with 8 slots
##   .. ..@ docvars :'data.frame':  8 obs. of  5 variables:
##   .. .. ..$ docname_: chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab
2015.txt" "Lab 2017.txt" ...
##   .. .. ..$ docid_  : Factor w/ 8 levels "Cons 2015.txt",..: 1 2 3 4 5 6 7
8
```

```
##   .. .. ..$ segid_  : int [1:8] 1 1 1 1 1 1 1 1
##   .. .. ..$ Party   : chr [1:8] "Cons" "Cons" "Lab" "Lab" ...
##   .. .. ..$ Year    : int [1:8] 2015 2017 2015 2017 2015 2017 2015 2017
##   .. ..@ meta    :List of 3
##   .. .. ..$ system:List of 5
##   .. .. .. ..$ package-version:Classes 'package_version',
'numeric_version'  hidden list of 1
##   .. .. .. .. ..$ : int [1:3] 3 3 1
##   .. .. .. ..$ r-version      :Classes 'R_system_version',
'package_version', 'numeric_version'  hidden list of 1
##   .. .. .. .. ..$ : int [1:3] 4 3 1
##   .. .. .. ..$ system         : Named chr [1:3] "Windows" "x86-64" "Miras"
##   .. .. .. .. ..- attr(*, "names")= chr [1:3] "sysname" "machine" "user"
##   .. .. .. ..$ directory      : chr "C:/Users/Miras/Desktop/u Milan/1st
year classes/Big Data Analystics/Labs/Lab1"
##   .. .. .. ..$ created        : Date[1:1], format: "2023-10-11"
##   .. .. ..$ object:List of 9
##   .. .. .. ..$ unit        : chr "documents"
##   .. .. .. ..$ what        : chr "word"
##   .. .. .. ..$ ngram       : int 1
##   .. .. .. ..$ skip        : int 0
##   .. .. .. ..$ concatenator: chr "_"
##   .. .. .. ..$ weight_tf   :List of 3
##   .. .. .. .. ..$ scheme: chr "count"
##   .. .. .. .. ..$ base  : NULL
##   .. .. .. .. ..$ k     : NULL
##   .. .. .. ..$ weight_df   :List of 5
##   .. .. .. .. ..$ scheme   : chr "unary"
##   .. .. .. .. ..$ base     : NULL
##   .. .. .. .. ..$ c        : NULL
##   .. .. .. .. ..$ smoothing: NULL
##   .. .. .. .. ..$ threshold: NULL
##   .. .. .. ..$ smooth      : num 0
##   .. .. .. ..$ summary     :List of 2
##   .. .. .. .. ..$ hash: chr(0)
##   .. .. .. .. ..$ data: NULL
##   .. .. ..$ user  : list()
##   .. ..@ i       : int [1:20687] 0 1 0 1 2 3 4 5 6 7 ...
##   .. ..@ p       : int [1:6214] 0 2 10 18 19 27 35 36 44 47 ...
##   .. ..@ Dim     : int [1:2] 8 6213
##   .. ..@ Dimnames:List of 2
##   .. .. ..$ docs    : chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab
2015.txt" "Lab 2017.txt" ...
##   .. .. ..$ features: chr [1:6213] "conservative" "parti" "manifesto"
"stong" ...
##   .. ..@ x       : num [1:20687] 2 1 17 11 13 19 27 16 33 42 ...
##   .. ..@ factors : list()
##  $ docs      : chr [1:8] "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt"
"Lab 2017.txt" ...
##  $ features  : chr [1:6213] "conservative" "parti" "manifesto" "stong" ...
```

```
##  $ dir       : num [1:2] 3 7
##  $ dispersion: chr "poisson"
##  $ priors    : num [1:4] Inf Inf 3 1
##  $ theta     : num [1:8] -0.0899 -0.1487 -0.4098 -0.5603 -0.9455 ...
##  $ beta      : num [1:6213] -0.1309 0.3341 0.4219 -0.0981 -0.4291 ...
##  $ psi       : num [1:6213] -1.01 3.03 2.16 -2.1 1.8 ...
##  $ alpha     : num [1:8] 0.1918 0.1829 -0.3392 -0.0512 0.2394 ...
##  $ phi       : num [1:6213] 1 1 1 1 1 1 1 1 1 1 ...
##  $ se.theta  : num [1:8] 0.0199 0.0197 0.0232 0.0182 0.0104 ...
##  $ call      : language textmodel_wordfish.dfm(x = myDfm, dir = c(3, 7))
##  - attr(*, "class")= chr [1:3] "textmodel_wordfish" "textmodel" "list"
```

```
scores_words <-data.frame(wfm$features, wfm$beta, wfm$psi)
str(scores_words)
```

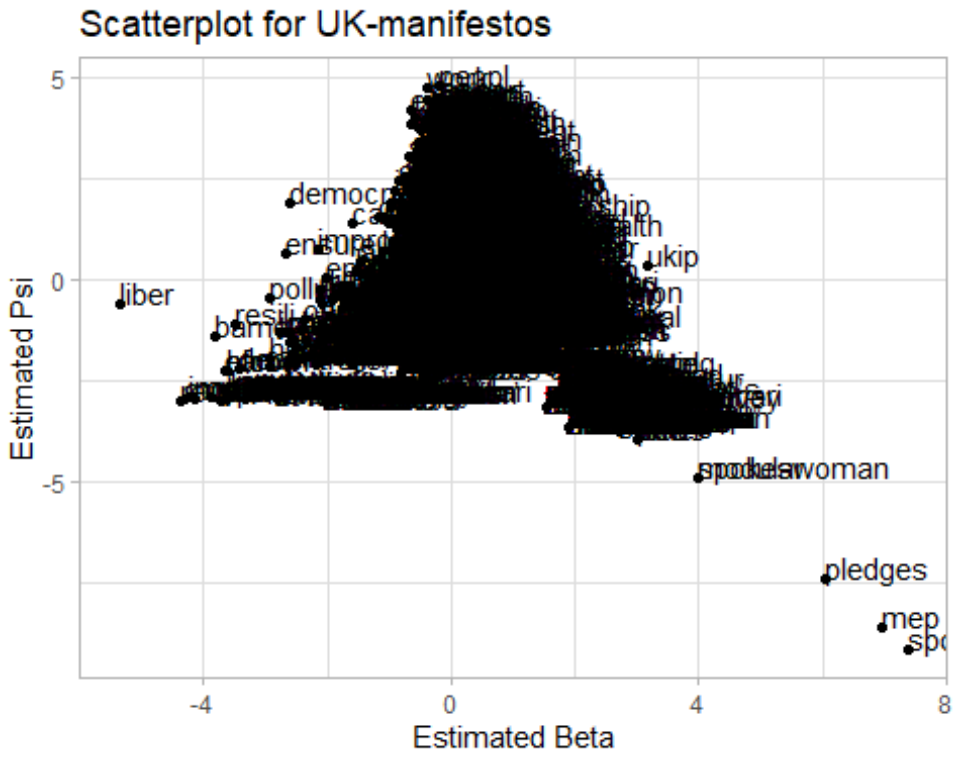
```
## 'data.frame':    6213 obs. of  3 variables:
##  $ wfm.features: chr  "conservative" "parti" "manifesto" "stong" ...
##  $ wfm.beta    : num  -0.1309 0.3341 0.4219 -0.0981 -0.4291 ...
##  $ wfm.psi     : num  -1.01 3.03 2.16 -2.1 1.8 ...
```

```
# Let's check for the correlation between psi and beta i.e. level of
idiosyncrasy at the level of the word and words differentiating document's
position along latent dimension
cor(abs(scores_words$wfm.beta), scores_words$wfm.psi)
```

```
## [1] -0.6959964
```

```
#negative correlation where psi is larger than beta
```

```
# Plot estimated word positions
ggplot(scores_words, aes(wfm.beta, wfm.psi, label= wfm.features))+
  geom_point() +geom_text(hjust=0, vjust=0) +
  theme_light() +
  labs(title  = "Scatterplot for UK-manifestos",
       x = "Estimated Beta",
       y = "Estimated Psi")
```



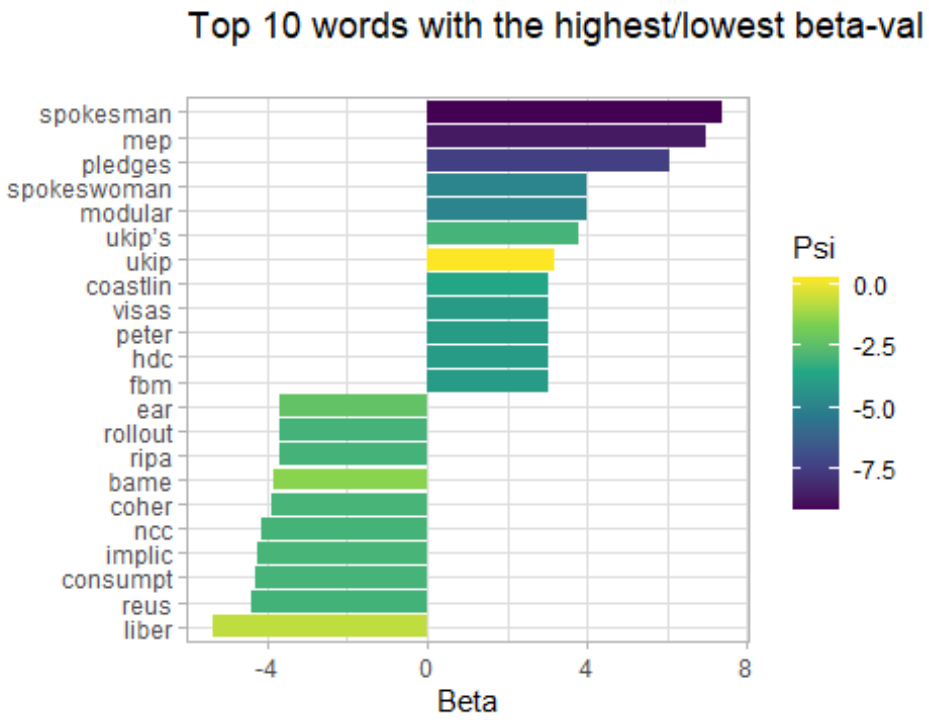

```
# top 40 features for negative beta: liber, democrat, pollut...
head(scores_words[order(scores_words$wfm.beta),], 40)
```

```
##      wfm.features  wfm.beta    wfm.psi
## 1131        liber -5.366044 -0.6311101
## 4406         reus -4.375078 -3.0249425
## 4242     consumpt -4.307773 -2.9665311
## 4281       implic -4.241997 -2.9095994
## 4240          ncc -4.140646 -2.9763313
## 4401        coher -3.870851 -2.9278937
## 3851         bame -3.815367 -1.3992096
## 4277      rollout -3.705023 -3.0107384
## 4538         ripa -3.705023 -3.0107384
## 3871          ear -3.662855 -2.2821166
## 4745           1p -3.662136 -2.7515164
## 4542      journal -3.627093 -2.9452451
## 4708       timber -3.627093 -2.9452451
## 3952        blood -3.615608 -2.2424710
## 4085        ofcom -3.615608 -2.2424710
## 4196   coalition’ -3.551024 -2.8816014
## 4246       reopen -3.551024 -2.8816014
## 4356         cell -3.551024 -2.8816014
## 3478       resili -3.491006 -1.1268454
## 4227     ringfenc -3.476749 -2.8197417
## 3750  decentralis -3.437603 -2.2276385
## 4223   decarbonis -3.318357 -2.9764868
## 4273     deforest -3.318357 -2.9764868
```

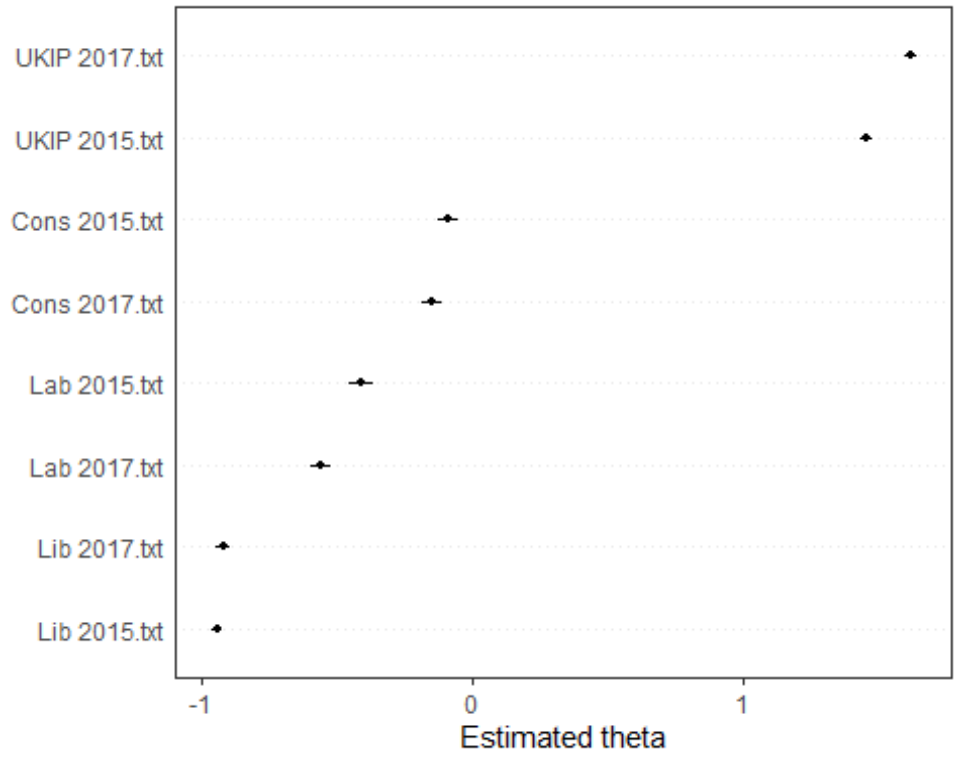
```
## 4291       refund -3.318357 -2.9764868
## 4340   standardis -3.318357 -2.9764868
## 4345         500m -3.318357 -2.9764868
## 4346         400m -3.318357 -2.9764868
## 4429      drainag -3.318357 -2.9764868
## 4459    liberalis -3.318357 -2.9764868
## 4509          wed -3.318357 -2.9764868
## 4543     authoris -3.318357 -2.9764868
## 4237     circular -3.235612 -2.9086442
## 4342  underrepres -3.235612 -2.9086442
## 4364     prescrib -3.235612 -2.9086442
## 4381       assert -3.235612 -2.9086442
## 4386           5p -3.235612 -2.9086442
## 4397       pollin -3.235612 -2.9086442
## 4445   driverless -3.235612 -2.9086442
## 4467       deviat -3.235612 -2.9086442
## 4557      encrypt -3.235612 -2.9086442

# top 40 words for positive beta: spokesman, pledges, spokeswoman...
tail(scores_words[order(scores_words$wfm.beta),], 40)

##      wfm.features wfm.beta    wfm.psi
## 5166        petit 2.562889 -2.7214972
## 5476  multicultur 2.562889 -2.7214972
## 5267         toni 2.655592 -3.0158736
## 5474         inde 2.655592 -3.0158736
## 5619         paul 2.701771 -3.7791012
## 5620       nuttal 2.701771 -3.7791012
## 5630       suzann 2.701771 -3.7791012
## 5631         evan 2.701771 -3.7791012
## 5638      concess 2.701771 -3.7791012
## 5654         tune 2.701771 -3.7791012
## 5666         mike 2.701771 -3.7791012
## 5667       hookem 2.701771 -3.7791012
## 5682     exclusiv 2.701771 -3.7791012
## 5691      patrick 2.701771 -3.7791012
## 5692      o'flynn 2.701771 -3.7791012
## 5728       gordon 2.701771 -3.7791012
## 5762     catalogu 2.701771 -3.7791012
## 5789       syndic 2.701771 -3.7791012
## 5794        circl 2.701771 -3.7791012
## 5795       miseri 2.701771 -3.7791012
## 5802     syllabus 2.701771 -3.7791012
## 5861         cllr 2.701771 -3.7791012
## 5863         pips 2.701771 -3.7791012
## 5875       blair' 2.701771 -3.7791012
## 5923        niqab 2.701771 -3.7791012
## 5936      dismiss 2.701771 -3.7791012
## 6073         bean 2.701771 -3.7791012
## 5352          cfp 2.889972 -2.9674135
```

```
## 5745          fbm 3.022315 -3.9811529
## 5750          hdc 3.022315 -3.9811529
## 5876        peter 3.022315 -3.9811529
## 5890        visas 3.022315 -3.9811529
## 5368     coastlin 3.039963 -3.6027874
## 2387         ukip 3.162109  0.3202199
## 5009       ukip’s 3.788465 -2.9682931
## 5735  spokeswoman 3.979006 -4.9021517
## 5744      modular 3.979006 -4.9021517
## 5731      pledges 6.041917 -7.4364882
## 5621          mep 6.965388 -8.6384593
## 5642    spokesman 7.370611 -9.1722136

# in this case we have just 6 documents and it's not very clear the meaning
of the latent dimension just
# by looking at betas (at least the first 40 features). Perhaps liberal vs.
conservative?

# let's extract the top 10 words with either the largest positive or negative
beta
scores_words2 <- top_n(scores_words, 10, wfm.beta )
scores_words2

##    wfm.features wfm.beta    wfm.psi
## 1          ukip 3.162109  0.3202199
## 2        ukip’s 3.788465 -2.9682931
## 3      coastlin 3.039963 -3.6027874
## 4           mep 6.965388 -8.6384593
## 5     spokesman 7.370611 -9.1722136
## 6       pledges 6.041917 -7.4364882
## 7   spokeswoman 3.979006 -4.9021517
## 8       modular 3.979006 -4.9021517
## 9           fbm 3.022315 -3.9811529
## 10          hdc 3.022315 -3.9811529
## 11        peter 3.022315 -3.9811529
## 12        visas 3.022315 -3.9811529

scores_words3 <- top_n(scores_words, -10, wfm.beta )
scores_words3

##    wfm.features  wfm.beta    wfm.psi
## 1         liber -5.366044 -0.6311101
## 2          bame -3.815367 -1.3992096
## 3           ear -3.662855 -2.2821166
## 4           ncc -4.140646 -2.9763313
## 5      consumpt -4.307773 -2.9665311
## 6       rollout -3.705023 -3.0107384
## 7        implic -4.241997 -2.9095994
## 8         coher -3.870851 -2.9278937
## 9          reus -4.375078 -3.0249425
## 10         ripa -3.705023 -3.0107384
```

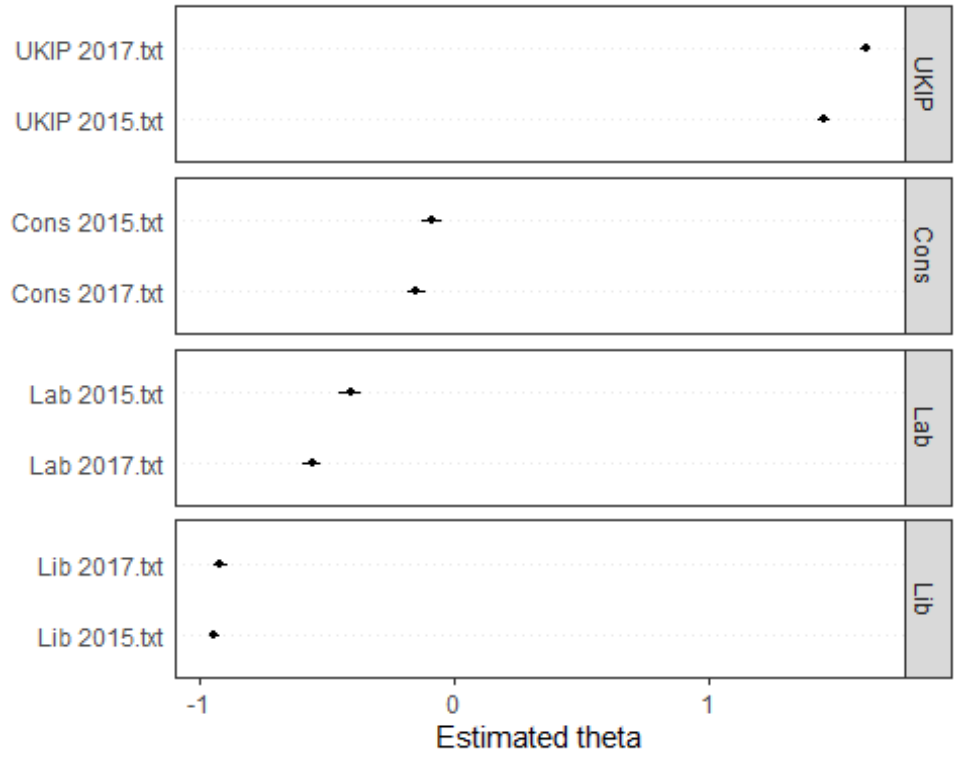
```r
scores_words_new <- rbind(scores_words2, scores_words3)
# reorder the features
scores_words_new  <-  mutate(scores_words_new, Feature= reorder(wfm.features,
wfm.beta))

ggplot(scores_words_new , aes(Feature, wfm.beta))  +
  geom_col(aes(fill = wfm.psi)) +
  scale_fill_viridis_c() +
  coord_flip() +
  theme_light() +
  labs(title = "\nTop 10 words with the highest/lowest beta-value\n",
       x = "",
       y = "Beta",
       fill = "Psi")
```



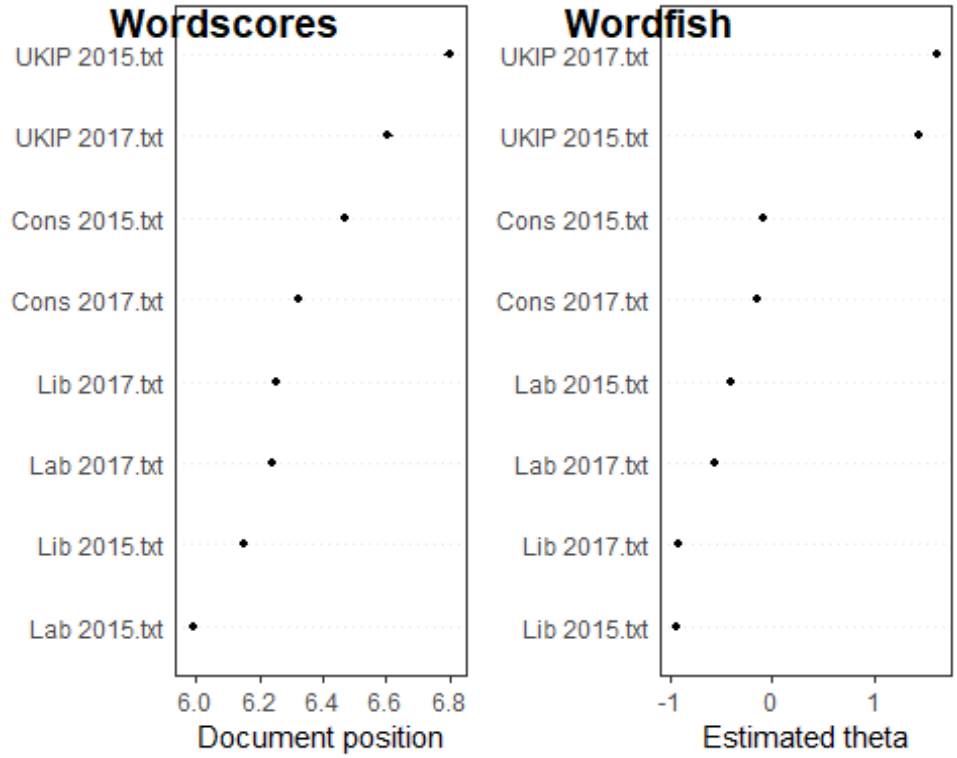

```r
#top 10 words that determine a document's position along latent dimension

textplot_scale1d(wfm)
```




```
textplot_scale1d(wfm,  groups = docvars(testCorpus, "Party"))
```



```
# Comparison of the results we got from Wordfish with the raw score ones we
got from Wordscores using the economic policy position

ws <- textmodel_wordscores(myDfm, c(7.85, NA, 3.85, NA, 5.14, NA, 8.57, NA))
pr_all <- predict(ws, interval = "confidence")

## Warning: 1456 features in newdata not used in prediction.

# Comparing wordscores vs wordfish
wordscores <- textplot_scale1d(pr_all)
wordfish <- textplot_scale1d(wfm)
plot_grid(wordscores , wordfish , labels = c('Wordscores', 'Wordfish'))
```




```
# check for the correlation
party <- wfm$docs
score_wf <-wfm$theta
score_ws <- pr_all$fit

scores_texts <-data.frame(party, score_wf, score_ws)
str(scores_texts)

## 'data.frame':    8 obs. of  5 variables:
##  $ party   : chr  "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##  $ score_wf: num  -0.0899 -0.1487 -0.4098 -0.5603 -0.9455 ...
##  $ fit     : num  6.47 6.32 5.99 6.24 6.15 ...
```

```
##  $ lwr     : num  6.46 6.31 5.98 6.22 6.14 ...
##  $ upr     : num  6.48 6.33 6 6.25 6.16 ...
```

```
colnames(scores_texts)[3] <- "score_ws"
str(scores_texts)
```

```
## 'data.frame':    8 obs. of  5 variables:
##  $ party   : chr  "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##  $ score_wf: num  -0.0899 -0.1487 -0.4098 -0.5603 -0.9455 ...
##  $ score_ws: num  6.47 6.32 5.99 6.24 6.15 ...
##  $ lwr     : num  6.46 6.31 5.98 6.22 6.14 ...
##  $ upr     : num  6.48 6.33 6 6.25 6.16 ...
```

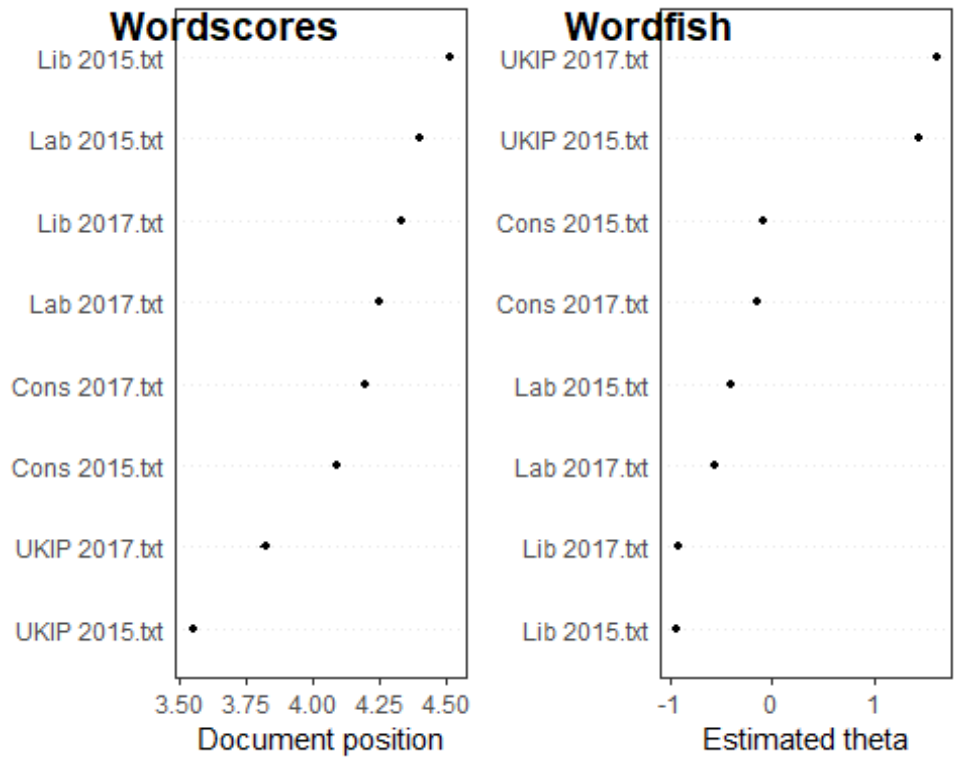
```
# high but not perfect correlation. Two different dimensions?
cor(scores_texts$score_ws, scores_texts$score_wf)
```

```
## [1] 0.852085
```

```
# Comparison of the results we got from Wordfish with the raw score ones we
got from Wordscores using the EU policy position

# Wordscores
ws2 <- textmodel_wordscores(myDfm, c(3.14, NA, 5.57, NA, 6.71, NA, 1.14, NA))
pr_all2 <- predict(ws2, interval = "confidence")
```

```
## Warning: 1456 features in newdata not used in prediction.
```

```
# Comparing wordscores vs wordfish
wordscores2 <- textplot_scale1d(pr_all2)
wordfish <- textplot_scale1d(wfm)
plot_grid(wordscores2 , wordfish , labels = c('Wordscores', 'Wordfish'))
```




```
party <- wfm$docs
score_wf <-wfm$theta
score_ws2 <- pr_all2$fit

scores_texts <-data.frame(party, score_wf, score_ws2)
str(scores_texts)

## 'data.frame':    8 obs. of  5 variables:
##  $ party   : chr  "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##  $ score_wf: num  -0.0899 -0.1487 -0.4098 -0.5603 -0.9455 ...
##  $ fit     : num  4.09 4.19 4.4 4.25 4.52 ...
##  $ lwr     : num  4.08 4.18 4.39 4.24 4.51 ...
##  $ upr     : num  4.1 4.2 4.41 4.27 4.53 ...

colnames(scores_texts)[3] <- "score_ws2"
str(scores_texts)

## 'data.frame':    8 obs. of  5 variables:
##  $ party    : chr  "Cons 2015.txt" "Cons 2017.txt" "Lab 2015.txt" "Lab
2017.txt" ...
##  $ score_wf : num  -0.0899 -0.1487 -0.4098 -0.5603 -0.9455 ...
##  $ score_ws2: num  4.09 4.19 4.4 4.25 4.52 ...
##  $ lwr      : num  4.08 4.18 4.39 4.24 4.51 ...
##  $ upr      : num  4.1 4.2 4.41 4.27 4.53 ...
```

```r
# high negative correlation. Two different dimensions for sure?
cor(scores_texts$score_ws2, scores_texts$score_wf)
```

```
## [1] -0.9314948
```