Package 'LSX'

September 22, 2020

Type Package

Title Model for Semisupervised Text Analysis Based on Word Embeddings

Date 2020-09-22

Version 0.9.2

Description A word embeddings-based semisupervised models for document scaling Watanabe (2017) <doi:10.1177/0267323117695735>.
LSS allows users to analyze large and complex corpora on arbitrary dimensions with seed words exploiting efficiency of word embeddings (SVD, Glove).

License GPL-3

LazyData TRUE

Encoding UTF-8

Depends quanteda (>= 2.0), quanteda.textmodels, methods, R (>= 3.5.0)

Imports digest, Matrix, RSpectra, irlba, rsvd, rsparse, proxyC, grDevices, stats, ggplot2, ggrepel, reshape2, e1071

Suggests testthat

RoxygenNote 7.1.1

NeedsCompilation no

Author Kohei Watanabe [aut, cre, cph]

Maintainer Kohei Watanabe <watanabe.kohei@gmail.com>

Repository CRAN

Date/Publication 2020-09-22 11:20:03 UTC

R topics documented:

`as.seedwords` *Convinient function to convert a list to seed words*

Description

Convinient function to convert a list to seed words

Usage

```
as.seedwords(x, upper = 1, lower = 2)
```

Arguments

`x`	a list of characters vectors or a dictionary object
`upper`	numeric index or key for seed words for higher scores
`lower`	numeric index or key for seed words for lower scores

Value

named numeric vector for seed words with polarity scores

`char_keyness` *Identify context words using user-provided patterns*

Description

Identify context words using user-provided patterns

Usage

```
char_keyness(
  x,
  pattern,
  valuetype = c("glob", "regex", "fixed"),
  case_insensitive = TRUE,
  window = 10,
  p = 0.001,
  min_count = 10,
  remove_pattern = TRUE,
  ...
)
```

Arguments

x	a tokens object created by quanteda::tokens().
pattern	quanteda::pattern() to specify target words
valuetype	the type of pattern matching: "glob" for "glob"-style wildcard expressions; "regex" for regular expressions; or "fixed" for exact matching. See quanteda::valuetype() for details.
case_insensitive	ignore case when matching, if TRUE
window	size of window for collocation analysis.
p	threshold for statistical significance of collocations.
min_count	minimum frequency of words within the window to be considered as collocations.
remove_pattern	if TRUE, keywords do not containe target words.
...	additional arguments passed to textstat_keyness().

See Also

tokens_select() and textstat_keyness()

Examples

```
require(quanteda)
con <- url("https://bit.ly/2GZwLcN", "rb")
corp <- readRDS(con)
close(con)
corp <- corpus_reshape(corp, 'sentences')
toks <- tokens(corp, remove_punct = TRUE)
toks <- tokens_remove(toks, stopwords())

# economy keywords
eco <- char_keyness(toks, 'econom*')
head(eco, 20)
```

```
# politics keywords
pol <- char_keyness(toks, 'politi*')
head(pol, 20)
```

`cohesion` *Computes cohesion of components of latent semantic analysis*

Description

Computes cohesion of components of latent semantic analysis

Usage

```
cohesion(object, bandwidth = 10)
```

Arguments

`object`	a fitted `textmodel_lss`
`bandwidth`	size of window for smoothing

`data_dictionary_ideology` *Seed words for analysis of left-right political ideology*

Description

Seed words for analysis of left-right political ideology

Examples

```
as.seedwords(data_dictionary_ideology)
```

`data_dictionary_sentiment`
Seed words for analysis of positive-negative sentiment

Description

Seed words for analysis of positive-negative sentiment

References

Turney, P. D., & Littman, M. L. (2003). Measuring Praise and Criticism: Inference of Semantic Orientation from Association. ACM Trans. Inf. Syst., 21(4), 315–346. https://doi.org/10.1145/944012.944013

Examples

```
as.seedwords(data_dictionary_sentiment)
```

`data_textmodel_lss_russianprotests`
A fitted LSS model on street protest in Russia

Description

This model was trained on a Russian media corpus (newspapers, TV transcripts and newswires) to analyze framing of street protests. The scale is protests as "freedom of expression" (high) vs "social disorder" (low). Although some slots are missing in this object (because the model was imported from the original Python implementation), it allows you to scale texts using `predict`.

References

Lankina, Tomila, and Kohei Watanabe. "'Russian Spring' or 'Spring Betrayal'? The Media as a Mirror of Putin's Evolving Strategy in Ukraine." Europe-Asia Studies 69, no. 10 (2017): 1526–56. https://doi.org/10.1080/09668136.2017.1397603.

diagnosys *Identify noisy documents in a corpus*

Description

Identify noisy documents in a corpus

Usage

```
diagnosys(x, ...)
```

Arguments

x	character or corpus object whose texts will be diagnosed
...	extra arguments passed to tokens

seedwords *Seed words for Latent Semantci Analysis*

Description

Seed words for Latent Semantci Analysis

Usage

```
seedwords(type)
```

Arguments

type	type of seed words currently only for sentiment (sentiment) or political ideology (ideology).

References

Turney, P. D., & Littman, M. L. (2003). Measuring Praise and Criticism: Inference of Semantic Orientation from Association. ACM Trans. Inf. Syst., 21(4), 315–346. https://doi.org/10.1145/944012.944013

Examples

```
seedwords('sentiment')
```

`smooth_lss` *Smooth predicted LSS scores by local polynomial regression*

Description

Smooth predicted LSS scores by local polynomial regression

Usage

```
smooth_lss(
  x,
  lss_var = "fit",
  date_var = "date",
  span = 0.1,
  from = NULL,
  to = NULL,
  ...
)
```

Arguments

`x`	a `data.frame` containing variables for LSS scores and dates
`lss_var`	the name of the column for LSS scores
`date_var`	the name of the columns for dates
`span`	determines the level of smoothing
`from`	start of the time period
`to`	end of the time period
`...`	extra arguments passed to `loess()`

`textmodel_lss` *A word embeddings-based semisupervised model for document scaling*

Description

A word embeddings-based semisupervised model for document scaling

Usage

```
textmodel_lss(x, ...)

## S3 method for class 'dfm'
textmodel_lss(
  x,
  seeds,
  terms = NULL,
  k = 300,
  slice = NULL,
  weight = "count",
  cache = FALSE,
  simil_method = "cosine",
  engine = c("RSpectra", "irlba", "rsvd"),
  include_data = FALSE,
  verbose = FALSE,
  ...
)

## S3 method for class 'fcm'
textmodel_lss(
  x,
  seeds,
  terms = NULL,
  w = 50,
  weight = "count",
  cache = FALSE,
  simil_method = "cosine",
  engine = c("rsparse"),
  verbose = FALSE,
  ...
)
```

Arguments

x	a dfm or fcm created by quanteda::dfm() or quanteda::fcm()
...	additional argument passed to the SVD engine
seeds	a character vector, named numeric vector or dictionary that contains seed words.
terms	words weighted as model terms. All the features of quanteda::dfm() or quanteda::fcm() will be used if not specified.
k	the number of singular values requested to the SVD engine. Only used when x is a dfm.
slice	a number or indices of the components of word vectors used to compute similarity; slice < k to truncate word vectors; useful for diagnosys and simulation.
weight	weighting scheme passed to quanteda::dfm_weight(). Ignored when engine is "rsparse".

cache	if TRUE, save retult of SVD for next execution with identical x and settings. Use the base::options(lss_cache_dir) to change the location cache files to be save.
simil_method	specifies method to compute similarity between features. The value is passed to quanteda::textstat_simil(), "cosine" is used otherwise.
engine	choose SVD engine between RSpectra::svds(), irlba::irlba(), and rsparse::GloVe().
include_data	if TRUE, fitted model include the dfm supplied as x.
verbose	show messages if TRUE.
w	the size of word vectors. Only used when x is a fcm

References

Watanabe, Kohei. "Measuring News Bias: Russia's Official News Agency ITAR-TASS' Coverage of the Ukraine Crisis." European Journal of Communication 32, no. 3 (March 20, 2017): 224–41. https://doi.org/10.1177/0267323117695735.

Examples

```
require(quanteda)
con <- url("https://bit.ly/2GZwLcN", "rb")
corp <- readRDS(con)
close(con)
toks <- corpus_reshape(corp, "sentences") %>%
        tokens(remove_punct = TRUE) %>%
        tokens_remove(stopwords("en")) %>%
        tokens_select("^[\\p{L}]+$", valuetype = "regex", padding = TRUE)
dfmt <- dfm(toks) %>%
        dfm_trim(min_termfreq = 10)

seed <- as.seedwords(data_dictionary_sentiment)

# SVD
lss_svd <- textmodel_lss(dfmt, seed)
summary(lss_svd)

# sentiment model on economy
eco <- head(char_keyness(toks, 'econom*'), 500)
svd_eco <- textmodel_lss(dfmt, seed, terms = eco)

# sentiment model on politics
pol <- head(char_keyness(toks, 'politi*'), 500)
svd_pol <- textmodel_lss(dfmt, seed, terms = pol)

# GloVe
fcmt  <- fcm(toks, context = "window", count = "weighted", weights = 1 / (1:5), tri = TRUE)
lss_glov <- textmodel_lss(fcmt, seed)
summary(lss_glov)
```

textplot_factor *Plot factors of latent semantic space*

Description

Plot factors of latent semantic space

Usage

```
textplot_factor(x, sort = TRUE)
```

Arguments

x	fitted textmodel_lss object
sort	sort factors by relevance if TRUE

textplot_simil *Plot similarity between seed words*

Description

Plot similarity between seed words

Usage

```
textplot_simil(x, group = FALSE)
```

Arguments

x	fitted textmodel_lss object
group	if TRUE group seed words by seed patterns and show average similarity

textplot_terms *Plot polarity scores of words*

Description

Plot polarity scores of words

Usage

```
textplot_terms(x, highlighted = NULL)
```

Arguments

x	fitted textmodel_lss object
highlighted	a character vector to specify words to be highlighted

Index