

# Package ‘seededlda’

January 7, 2022

**Type** Package

**Title** Seeded-LDA for Topic Modeling

**Version** 0.8.0

**Description** Implements the seeded-LDA model (Lu, Ott, Cardie & Tsou 2010) <[doi:10.1109/ICDMW.2011.125](https://doi.org/10.1109/ICDMW.2011.125)> using the quanteda package and the GibbsLDA++ library for semisupervised topic modeling. Seeded-LDA allows users to pre-define topics with keywords to perform theory-driven analysis of textual data in social sciences and humanities (Watanabe & Zhou 2020) <[doi:10.1177/0894439320907027](https://doi.org/10.1177/0894439320907027)>.

**License** GPL-3

**URL** <https://github.com/koheiw/seededlda>

**BugReports** <https://github.com/koheiw/seededlda/issues>

**Encoding** UTF-8

**Depends** R (>= 3.5.0), quanteda (> 2.0), methods

**Imports** Matrix

**LinkingTo** Rcpp, RcppParallel, RcppArmadillo (>= 0.7.600.1.0), quanteda

**Suggests** testthat, quanteda.textmodels, topicmodels

**RoxygenNote** 7.1.2

**NeedsCompilation** yes

**Author** Kohei Watanabe [aut, cre, cph],  
Phan Xuan-Hieu [aut, cph] (GibbsLDA++)

**Maintainer** Kohei Watanabe <[watanabe.kohei@gmail.com](mailto:watanabe.kohei@gmail.com)>

**Repository** CRAN

**Date/Publication** 2022-01-07 11:50:01 UTC

## R topics documented:

terms	2
textmodel_lda	2
topics	4

**Index****6**


---

terms	<i>Extract most likely terms</i>
-------	----------------------------------

---

**Description**

terms() returns the most likely terms, or words, for topics based on the phi parameter.

**Usage**

```
terms(x, n = 10)
```

**Arguments**

x	a LDA model fitted by <code>textmodel_seededlda()</code> or <code>textmodel_lda()</code>
n	number of terms to be extracted

**Details**

Users can access the original matrix `x$phi` for likelihood scores.

---

textmodel_lda	<i>Semisupervised Latent Dirichlet allocation</i>
---------------	---

---

**Description**

textmodel\_seededlda() implements semisupervised Latent Dirichlet allocation (seeded-LDA). The estimator's code adopted from the GibbsLDA++ library (Xuan-Hieu Phan, 2007). textmodel\_seededlda() allows identification of pre-defined topics by semisupervised learning with a seed word dictionary.

**Usage**

```
textmodel_lda(
  x,
  k = 10,
  max_iter = 2000,
  alpha = NULL,
  beta = NULL,
  model = NULL,
  verbose = quanteda_options("verbose")
)

textmodel_seededlda(
  x,
  dictionary,
```

```

    valuetype = c("glob", "regex", "fixed"),
    case_insensitive = TRUE,
    residual = FALSE,
    weight = 0.01,
    max_iter = 2000,
    alpha = NULL,
    beta = NULL,
    ...,
    verbose = quanteda_options("verbose")
)

```

### Arguments

x	the dfm on which the model will be fit
k	the number of topics; determined automatically by the number of keys in dictionary in <code>textmodel_seededlda()</code> .
max_iter	the maximum number of iteration in Gibbs sampling.
alpha	the value to smooth topic-document distribution; defaults to $\alpha = 50 / k$ .
beta	the value to smooth topic-word distribution; defaults to $\beta = 0.1$ .
model	a fitted LDA model; if provided, <code>textmodel_lda()</code> inherits parameters from an existing model. See details.
verbose	logical; if TRUE print diagnostic information during fitting.
dictionary	a <code>quanteda::dictionary()</code> with seed words that define topics.
valuetype	see <code>quanteda::valuetype</code>
case_insensitive	see <code>quanteda::valuetype</code>
residual	if TRUE a residual topic (or "garbage topic") will be added to user-defined topics.
weight	pseudo count given to seed words as a proportion of total number of words in x.
...	passed to <code>quanteda::dfm_trim</code> to restrict seed words based on their term or document frequency. This is useful when glob patterns in the dictionary match too many words.

### Details

To predict topics of new documents (i.e. out-of-sample), first, create a new LDA model from a existing LDA model passed to `model` in `textmodel_lda()`; second, apply `topics()` to the new model. The `model` argument takes objects created either by `textmodel_lda()` or `textmodel_seededlda()`.

### Value

`textmodel_seededlda()` and `textmodel_lda()` returns a list of model parameters. `theta` is the distribution of topics over documents; `phi` is the distribution of words over topics. `alpha` and `beta` are the small constant added to the frequency of words to estimate `theta` and `phi`, respectively, in Gibbs sampling. Other elements in the list subject to change.

## References

- Lu, Bin et al. (2011). "Multi-aspect Sentiment Analysis with Topic Models". doi:10.5555/2117693.2119585. *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*.
- Watanabe, Kohei & Zhou, Yuan (2020). "Theory-Driven Analysis of Large Corpora: Semisupervised Topic Classification of the UN Speeches". doi:10.1177/0894439320907027. *Social Science Computer Review*.

## See Also

[topicmodels](#)

## Examples

```
require(seededlda)
require(quanteda)

data("data_corpus_moviereviews", package = "quanteda.textmodels")
corp <- head(data_corpus_moviereviews, 500)
toks <- tokens(corp, remove_punct = TRUE, remove_symbols = TRUE, remove_number = TRUE)
dfmt <- dfm(toks) %>%
  dfm_remove(stopwords('en'), min_nchar = 2) %>%
  dfm_trim(min_termfreq = 0.90, termfreq_type = "quantile",
           max_docfreq = 0.1, docfreq_type = "prop")

# unsupervised LDA
lda <- textmodel_lda(head(dfmt, 450), 6)
terms(lda)
topics(lda)
lda2 <- textmodel_lda(tail(dfmt, 50), model = lda) # new documents
topics(lda2)

# semisupervised LDA
dict <- dictionary(list(people = c("family", "couple", "kids"),
                        space = c("alien", "planet", "space"),
                        moster = c("monster*", "ghost*", "zombie*"),
                        war = c("war", "soldier*", "tanks"),
                        crime = c("crime*", "murder", "killer")))
sllda <- textmodel_seededlda(dfmt, dict, residual = TRUE, min_termfreq = 10)
terms(sllda)
topics(sllda)
```

---

topics

*Extract most likely topics*

---

## Description

topics() returns the most likely topics for documents based on the theta parameter.

**Usage**

`topics(x)`

**Arguments**

`x` a LDA model fitted by `textmodel_seededlda()` or `textmodel_lda()`

**Details**

Users can access the original matrix `x$theta` for likelihood scores; run `max.col(x$theta)` to obtain the same result as `topics(x)`.

# Index

## \* **textmodel**

textmodel\_lda, 2

quanteda::dfm\_trim, 3

quanteda::dictionary(), 3

quanteda::valuetype, 3

terms, 2

textmodel\_lda, 2

textmodel\_lda(), 2, 5

textmodel\_seededlda(textmodel\_lda), 2

textmodel\_seededlda(), 2, 5

topicmodels, 4

topics, 4

topics(), 3