

# Package ‘keyATM’

July 29, 2020

**Version** 0.3.1

**Title** Keyword Assisted Topic Model

**Description**

Fits keyword assisted topic models (keyATM) using collapsed Gibbs samplers. The keyATM combines the latent dirichlet allocation (LDA) models with a small number of keywords selected by researchers in order to improve the interpretability and topic classification of the LDA. The key-ATM can also incorporate covariates and directly model time trends. The keyATM is proposed in Eshima, Imai, and Sasaki (2020) <arXiv:2004.05964>.

**License** GPL-3

**Depends** R (>= 3.6)

**Imports** Rcpp, dplyr (>= 1.0.0), fastmap, ggplot2, ggrepel, magrittr, Matrix, parallel, purrr, quanteda (>= 2.0.0), rlang, stats, stringr, tibble, tidyr (>= 1.0.0)

**LinkingTo** Rcpp, RcppEigen, RcppProgress

**Suggests** readtext, testthat (>= 2.1.0)

**URL** <https://keyatm.github.io/keyATM/>

**Encoding** UTF-8

**BugReports** <https://github.com/keyATM/keyATM/issues>

**LazyData** TRUE

**RoxygenNote** 7.1.1

**SystemRequirements** C++11

**NeedsCompilation** yes

**Author** Shusei Eshima [aut, cre] (<<https://orcid.org/0000-0003-3613-4046>>),  
Tomoya Sasaki [aut],  
William Lowe [ctb],  
Kosuke Imai [aut],  
Chung-hong Chan [ctb] (<<https://orcid.org/0000-0002-6232-7530>>),  
Romain François [ctb] (<<https://orcid.org/0000-0002-2444-4226>>)

**Maintainer** Shusei Eshima <[shuseieshima@g.harvard.edu](mailto:shuseieshima@g.harvard.edu)>

**Repository** CRAN

**Date/Publication** 2020-07-29 05:10:03 UTC

## R topics documented:

keyATM-package	2
by_strata_DocTopic	3
by_strata_TopicWord	3
covariates_get	4
covariates_info	4
keyATM	5
keyATMvb	7
keyATM_data_bills	9
keyATM_read	9
plot.strata_doctopic	10
plot_alpha	11
plot_modelfit	12
plot_pi	12
plot_timetrend	13
predict.keyATM_output	14
read_keywords	16
save.keyATM_output	17
save_fig	17
top_docs	18
top_topics	18
top_words	19
values_fig	19
visualize_keywords	20
weightedLDA	21
<b>Index</b>	<b>23</b>

---

keyATM-package	<i>Keyword Assisted Topic Models</i>
----------------	--------------------------------------

---

## Description

The implementation of keyATM models.

## Author(s)

**Maintainer:** Shusei Eshima <shuseieshima@g.harvard.edu> ([ORCID](#))

Authors:

- Tomoya Sasaki <tomoyas@mit.edu>
- Kosuke Imai <imai@harvard.edu>

Other contributors:

- William Lowe <wlowe@princeton.edu> [contributor]
- Chung-hong Chan <chainsawtiney@gmail.com> ([ORCID](#)) [contributor]
- Romain François ([ORCID](#)) [contributor]

**See Also**

Useful links:

- <https://keyatm.github.io/keyATM/>
- Report bugs at <https://github.com/keyATM/keyATM/issues>

---

by\_strata\_DocTopic     *Estimate document-topic distribution by strata (for covariate models)*

---

**Description**

Estimate document-topic distribution by strata (for covariate models)

**Usage**

```
by_strata_DocTopic(x, by_var, labels, by_values = NULL, ...)
```

**Arguments**

x	the output from the covariate keyATM model (see <a href="#">keyATM()</a> ).
by_var	character. The name of the variable to use.
labels	character. The labels for the values specified in by_var (ascending order).
by_values	numeric. Specific values for by_var, ordered from small to large. If it is not specified, all values in by_var will be used.
...	other arguments passed on to the <a href="#">predict()</a> function.

**Value**

strata\_topicword object (a list).

---

by\_strata\_TopicWord     *Estimate subsetted topic-word distribution*

---

**Description**

Estimate subsetted topic-word distribution

**Usage**

```
by_strata_TopicWord(x, keyATM_docs, by)
```

**Arguments**

- x                    the output from a keyATM model (see [keyATM\(\)](#)).
- keyATM\_docs        an object generated by [keyATM\\_read\(\)](#).
- by                   a vector whose length is the number of documents.

**Value**

strata\_topicword object (a list).

---

<code>covariates_get</code>	<i>Return covariates used in the iteration</i>
-----------------------------	--

---

**Description**

Return covariates used in the iteration

**Usage**

```
covariates_get(x)
```

**Arguments**

- x                    the output from the covariate keyATM model (see [keyATM\(\)](#))

---

<code>covariates_info</code>	<i>Show covariates information</i>
------------------------------	------------------------------------

---

**Description**

Show covariates information

**Usage**

```
covariates_info(x)
```

**Arguments**

- x                    the output from the covariate keyATM model (see [keyATM\(\)](#)).

---

keyATM	<i>keyATM main function</i>
--------	-----------------------------

---

## Description

Fit keyATM models.

## Usage

```
keyATM(
  docs,
  model,
  no_keyword_topics,
  keywords = list(),
  model_settings = list(),
  priors = list(),
  options = list(),
  keep = c()
)
```

## Arguments

docs	texts read via <a href="#">keyATM_read()</a> .
model	keyATM model: base, covariates, dynamic, and label.
no_keyword_topics	the number of regular topics.
keywords	a list of keywords.
model_settings	a list of model specific settings (details are in the online documentation).
priors	a list of priors of parameters.
options	a list of options <ul style="list-style-type: none"> <li>• <b>seed</b>: A numeric value for random seed. If it is not provided, the package randomly selects a seed.</li> <li>• <b>iterations</b>: An integer. Number of iterations. Default is 1500.</li> <li>• <b>verbose</b>: If TRUE, it prints loglikelihood and perplexity. Default is FALSE.</li> <li>• <b>llk_per</b>: An integer. If the value is <math>j</math> <b>keyATM</b> stores loglikelihood and perplexity every <math>j</math> iteration. Default value is 10 per iterations</li> <li>• <b>use_weights</b>: If TRUE use weight. Default is TRUE.</li> <li>• <b>weights_type</b>: There are four types of weights. Weights based on the information theory (<code>information-theory</code>) and inverse frequency (<code>inv-freq</code>) and normalized versions of them (<code>information-theory-normalized</code> and <code>inv-freq-normalized</code>). Default is <code>information-theory</code>.</li> <li>• <b>prune</b>: If TRUE rume keywords that do not appear in the corpus. Default is TRUE.</li> </ul>

- **store\_theta**: If TRUE or 1, it stores  $\theta$  (document-topic distribution) for the iteration specified by thinning. Default is FALSE (same as  $\emptyset$ ).
- **store\_pi**: If TRUE or 1, it stores  $\pi$  (the probability of using keyword topic word distribution) for the iteration specified by thinning. Default is FALSE (same as  $\emptyset$ ).
- **thinning**: An integer. If the value is  $j$  **keyATM** stores following parameters every  $j$  iteration. The default is 5.
  - *theta*: For all models. If store\_theta is TRUE document-level topic assignment is stored (sufficient statistics to calculate document-topic distributions theta).
  - *alpha*: For the base and dynamic models. In the base model alpha is shared across all documents whereas each state has different alpha in the dynamic model.
  - *lambda*: coefficients in the covariate model.
  - *R*: For the dynamic model. The state each document belongs to.
  - *P*: For the dynamic model. The state transition probability.
- **parallel\_init**: Parallelize processes to speed up initialization. Default is FALSE. Note that even if you use the same seed, the initialization will become different between with and without parallelization.

**keep** a vector of the names of elements you want to keep in output.

## Value

A keyATM\_output object containing:

**keyword\_k** number of keyword topics  
**no\_keyword\_topics** number of no-keyword topics  
**V** number of terms (number of unique words)  
**N** number of documents  
**model** the name of the model  
**theta** topic proportions for each document (document-topic distribution)  
**phi** topic specific word generation probabilities (topic-word distribution)  
**topic\_counts** number of tokens assigned to each topic  
**word\_counts** number of times each word type appears  
**doc\_lens** length of each document in tokens  
**vocab** words in the vocabulary (a vector of unique words)  
**priors** priors  
**options** options  
**keywords\_raw** specified keywords  
**model\_fit** perplexity and log-likelihood  
**pi** estimated  $\pi$  (the probability of using keyword topic word distribution) for the last iteration  
**values\_iter** values stored during iterations  
**kept\_values** outputs you specified to store in keep option  
**information** information about the fitting

**See Also**

`save.keyATM_output()`, [https://keyatm.github.io/keyATM/articles/pkgdown\\_files/Options.html](https://keyatm.github.io/keyATM/articles/pkgdown_files/Options.html)

**Examples**

```
## Not run:
library(keyATM)
library(quanteda)
data(keyATM_data_bills)
bills_keywords <- keyATM_data_bills$keywords
bills_dfm <- keyATM_data_bills$doc_dfm # quanteda dfm object
keyATM_docs <- keyATM_read(bills_dfm)

# keyATM Base
out <- keyATM(docs = keyATM_docs, model = "base",
             no_keyword_topics = 5, keywords = bills_keywords)

# keyATM Covariates
bills_cov <- as.data.frame(keyATM_data_bills$cov)
out <- keyATM(docs = keyATM_docs, model = "covariates",
             no_keyword_topics = 5, keywords = bills_keywords,
             model_settings = list(covariates_data = bills_cov,
                                   covariates_formula = ~ RepParty))

# keyATM Dynamic
bills_time_index <- keyATM_data_bills$time_index
# Time index should start from 1 and increase by 1
bills_time_index <- as.integer(bills_time_index - 100)
out <- keyATM(docs = keyATM_docs, model = "dynamic",
             no_keyword_topics = 5, keywords = bills_keywords,
             model_settings = list(num_states = 5,
                                   time_index = bills_time_index))

# Visit our website for full examples: https://keyatm.github.io/keyATM/

## End(Not run)
```

---

keyATMvb

*keyATM with Collapsed Variational Bayes*

---

**Description**

**Experimental feature:** Fit keyATM base with Collapsed Variational Bayes

## Usage

```
keyATMvb(  
  docs,  
  model,  
  no_keyword_topics,  
  keywords = list(),  
  model_settings = list(),  
  vb_options = list(),  
  priors = list(),  
  options = list(),  
  keep = list()  
)
```

## Arguments

docs	texts read via <a href="#">keyATM_read()</a>
model	keyATM model: base, covariates, and dynamic
no_keyword_topics	the number of regular topics
keywords	a list of keywords
model_settings	a list of model specific settings (details are in the online documentation)
vb_options	a list of settings for Variational Bayes <ul style="list-style-type: none"><li>• <b>convtol</b>: the default is 1e-4</li><li>• <b>init</b>: mcmc (default) or random</li></ul>
priors	a list of priors of parameters
options	a list of options same as <a href="#">keyATM()</a> . Options are used when initialization method is mcmc.
keep	a vector of the names of elements you want to keep in output

## Value

A keyATM\_output object

## See Also

[https://keyatm.github.io/keyATM/articles/pkgdown\\_files/keyATMvb.html](https://keyatm.github.io/keyATM/articles/pkgdown_files/keyATMvb.html)

---

keyATM_data_bills	<i>Bills data</i>
-------------------	-------------------

---

**Description**

Bills data

**Usage**

keyATM\_data\_bills

**Format**

A list with following objects:

**doc\_dfm** A quanteda dfm object of 140 documents. The text data is a part of the Congressional Bills scraped from <https://www.congress.gov>.

**cov** An integer vector which takes one if the Republican proposed the bill.

**keywords** A list of length 4 which contains keywords for four selected topics.

**time\_index** An integer vector indicating the session number of each bill.

**labels** An integer vector indicating 40 labels.

**labels\_all** An integer vector indicating all labels.

**Source**

<https://www.congress.gov>

---

keyATM_read	<i>Read texts</i>
-------------	-------------------

---

**Description**

Read texts and create a keyATM\_docs object, which is a list of texts.

**Usage**

```
keyATM_read(texts, encoding = "UTF-8", check = TRUE, progress_bar = FALSE)
```

**Arguments**

texts	input. keyATM takes quanteda dfm (dgCMatrix), data.frame, <b>tibble</b> tbl_df, or a vector of file paths.
encoding	character. Only used when texts is a vector of file paths. Default is UTF-8.
check	logical. If TRUE, check whether there is anything wrong with the structure of texts. Default is TRUE.
progress_bar	logical. If TRUE, it shows a progress bar (currently it only supports a quanteda object). Default is FALSE.

**Value**

a list whose elements are splitted texts. The length of the list equals to the number of documents.

**Examples**

```
## Not run:
# Use quanteda dfm
keyATM_docs <- keyATM_read(texts = quanteda_dfm)

# Use data.frame or tibble (texts should be stored in a column named `text`)
keyATM_docs <- keyATM_read(texts = data_frame_object)
keyATM_docs <- keyATM_read(texts = tibble_object)

# Use a vector that stores full paths to the text files
files <- list.files(doc_folder, pattern = "*.txt", full.names = TRUE)
keyATM_docs <- keyATM_read(texts = files)

## End(Not run)
```

---

plot.strata\_doctopic *Plot document-topic distribution by strata (for covariate models)*

---

**Description**

Plot document-topic distribution by strata (for covariate models)

**Usage**

```
## S3 method for class 'strata_doctopic'
plot(
  x,
  show_topic = NULL,
  var_name = NULL,
  by = c("topic", "covariate"),
  ci = 0.9,
  method = c("hdi", "eti"),
  point = c("mean", "median"),
  width = 0.1,
  show_point = TRUE,
  ...
)
```

**Arguments**

x a strata\_doctopic object (see [by\\_strata\\_DocTopic\(\)](#)).

show\_topic a vector or an integer. Indicate topics to visualize.

var_name	the name of the variable in the plot.
by	topic or covariate. Default is by topic.
ci	value of the credible interval (between 0 and 1) to be estimated. Default is 0.9 (90%).
method	method for computing the credible interval. The Highest Density Interval (hdi, default) or Equal-tailed Interval (eti).
point	method for computing the point estimate. mean (default) or median.
width	numeric. Width of the error bars.
show_point	logical. Show point estimates. The default is TRUE.
...	additional arguments not used.

**Value**

keyATM\_fig object.

**See Also**

[save\\_fig\(\)](#), [by\\_strata\\_DocTopic\(\)](#)

---

plot_alpha	<i>Show a diagnosis plot of alpha</i>
------------	---------------------------------------

---

**Description**

Show a diagnosis plot of alpha

**Usage**

```
plot_alpha(x, start = 0, show_topic = NULL, scales = "fixed")
```

**Arguments**

x	the output from a keyATM model (see <a href="#">keyATM()</a> ).
start	integer. The start of slice iteration. Default is 0.
show_topic	a vector to specify topic indexes to show. Default is NULL.
scales	character. Control the scale of y-axis (the parameter in <a href="#">ggplot2::facet_wrap()</a> ): free adjusts y-axis for parameters. Default is fixed.

**Value**

keyATM\_fig object

**See Also**

[save\\_fig\(\)](#)

---

plot_modelfit	<i>Show a diagnosis plot of log-likelihood and perplexity</i>
---------------	---

---

**Description**

Show a diagnosis plot of log-likelihood and perplexity

**Usage**

```
plot_modelfit(x, start = 1)
```

**Arguments**

`x` the output from a keyATM model (see [keyATM\(\)](#)).  
`start` integer. The starting value of iteration to use in plot. Default is 1.

**Value**

keyATM\_fig object.

**See Also**

[save\\_fig\(\)](#)

---

plot_pi	<i>Show a diagnosis plot of pi</i>
---------	------------------------------------

---

**Description**

Show a diagnosis plot of pi

**Usage**

```
plot_pi(  
  x,  
  show_topic = NULL,  
  start = 0,  
  ci = 0.9,  
  method = c("hdi", "eti"),  
  point = c("mean", "median")  
)
```

**Arguments**

x	the output from a keyATM model (see <a href="#">keyATM()</a> ).
show_topic	an integer or a vector. Indicate topics to visualize. Default is NULL.
start	integer. The starting value of iteration to use in the plot. Default is 0.
ci	value of the credible interval (between 0 and 1) to be estimated. Default is 0.9 (90%). This is an option when calculating credible intervals (you need to set store_pi = TRUE in <a href="#">keyATM()</a> ).
method	method for computing the credible interval. The Highest Density Interval (hdi, default) or Equal-tailed Interval (eti). This is an option when calculating credible intervals (you need to set store_pi = TRUE in <a href="#">keyATM()</a> ).
point	method for computing the point estimate. mean (default) or median. This is an option when calculating credible intervals (you need to set store_pi = TRUE in <a href="#">keyATM()</a> ).

**Value**

keyATM\_fig object.

**See Also**

[save\\_fig\(\)](#)

---

plot_timetrend	<i>Plot time trend</i>
----------------	------------------------

---

**Description**

Plot time trend

**Usage**

```
plot_timetrend(
  x,
  show_topic = NULL,
  time_index_label = NULL,
  ci = 0.9,
  method = c("hdi", "eti"),
  point = c("mean", "median"),
  xlab = "Time",
  scales = "fixed",
  width = 0.5,
  show_point = TRUE,
  ...
)
```

**Arguments**

x	the output from the dynamic keyATM model (see <a href="#">keyATM()</a> ).
show_topic	an integer or a vector. Indicate topics to visualize. Default is NULL.
time_index_label	a vector. The label for time index. The length should be equal to the number of documents (time index provided to <a href="#">keyATM()</a> ).
ci	value of the credible interval (between 0 and 1) to be estimated. Default is 0.9 (90%). This is an option when calculating credible intervals (you need to set store_theta = TRUE in <a href="#">keyATM()</a> ).
method	method for computing the credible interval. The Highest Density Interval (hdi, default) or Equal-tailed Interval (eti). This is an option when calculating credible intervals (you need to set store_theta = TRUE in <a href="#">keyATM()</a> ).
point	method for computing the point estimate. mean (default) or median. This is an option when calculating credible intervals (you need to set store_theta = TRUE in <a href="#">keyATM()</a> ).
xlab	a character.
scales	character. Control the scale of y-axis (the parameter in <a href="#">ggplot2::facet_wrap()</a> ): free adjusts y-axis for parameters. Default is fixed.
width	numeric. Width of the error bars.
show_point	logical. The default is TRUE. This is an option when calculating credible intervals.
...	additional arguments not used.

**Value**

keyATM\_fig object.

**See Also**

[save\\_fig\(\)](#)

---

predict.keyATM\_output *Predict topic proportions for the covariate keyATM*

---

**Description**

Predict topic proportions for the covariate keyATM

**Usage**

```
## S3 method for class 'keyATM_output'
predict(
  object,
  newdata,
  transform = FALSE,
  burn_in = NULL,
  parallel = TRUE,
  mc.cores = NULL,
  posterior_mean = TRUE,
  ci = 0.9,
  method = c("hdi", "eti"),
  point = c("mean", "median"),
  label = NULL,
  raw_values = FALSE,
  ...
)
```

**Arguments**

<code>object</code>	the <code>keyATM_output</code> object for the covariate model.
<code>newdata</code>	New observations which should be predicted.
<code>transform</code>	Transform and standardize the <code>newdata</code> with the same formula and option as <code>model_settings</code> used in <code>keyATM()</code> .
<code>burn_in</code>	integer. Burn-in period. If not specified, it is the half of samples. Default is <code>NULL</code> .
<code>parallel</code>	logical. If <code>TRUE</code> , parallelization for speeding up. Default is <code>TRUE</code> .
<code>mc.cores</code>	integer. The number of cores to use. Default is <code>NULL</code> .
<code>posterior_mean</code>	logical. If <code>TRUE</code> , the quantity of interest to estimate is the posterior mean. Default is <code>TRUE</code> .
<code>ci</code>	value of the credible interval (between 0 and 1) to be estimated. Default is 0.9 (90%).
<code>method</code>	method for computing the credible interval. The Highest Density Interval ( <code>hdi</code> , default) or Equal-tailed Interval ( <code>eti</code> ).
<code>point</code>	method for computing the point estimate. <code>mean</code> (default) or <code>median</code> .
<code>label</code>	a character. Add a <code>label</code> column to the output. The default is <code>NULL</code> (do not add it).
<code>raw_values</code>	a logical. Returns raw values. The default is <code>FALSE</code> .
<code>...</code>	additional arguments not used.

---

`read_keywords`*Convert a quanteda dictionary to keywords*

---

**Description**

This function converts or reads a dictionary object from quanteda to a named list. "Glob"-style wildcard expressions (e.g. `politic*`) are resolved based on the available terms in your texts.

**Usage**

```
read_keywords(file = NULL, docs = NULL, dictionary = NULL, split = TRUE, ...)
```

**Arguments**

<code>file</code>	file identifier for a foreign dictionary, e.g. path to a dictionary in YAML or LIWC format
<code>docs</code>	texts read via <code>keyATM_read()</code>
<code>dictionary</code>	a quanteda dictionary object, ignore if file is not NULL
<code>split</code>	boolean, if multi-word terms be seperated, e.g. "air force" splits into "air" and "force".
<code>...</code>	additional parameters for <code>quanteda::dictionary()</code>

**Value**

a named list which can be used as keywords for e.g. `keyATM()`

**See Also**

[dictionary](#)

**Examples**

```
## Not run:
library(keyATM)
library(quanteda)
## using the moral foundation dictiionary example from quanteda
dictfile <- tempfile()
download.file("http://bit.ly/37cV95h", dictfile)
data(keyATM_data_bills)
bills_dfm <- keyATM_data_bills$doc_dfm
keyATM_docs <- keyATM_read(bills_dfm)
read_keywords(file = dictfile, docs = keyATM_docs, format = "LIWC")

## End(Not run)
```

---

save.keyATM_output	<i>Save a keyATM_output object</i>
--------------------	------------------------------------

---

**Description**

Save a keyATM\_output object

**Usage**

```
save.keyATM_output(x, file = stop("'file' must be specified"))
```

**Arguments**

x	a keyATM_output object (see <a href="#">keyATM()</a> ).
file	file name to create on disk.

**See Also**

[keyATM\(\)](#), [weightedLDA\(\)](#), [keyATMvb\(\)](#)

---

save_fig	<i>Save a figure</i>
----------	----------------------

---

**Description**

Save a figure

**Usage**

```
save_fig(x, filename, ...)
```

**Arguments**

x	the keyATM_fig object.
filename	file name to create on disk.
...	other arguments passed on to the <a href="#">ggplot2::ggsave()</a> function.

**See Also**

[visualize\\_keywords\(\)](#), [plot\\_alpha\(\)](#), [plot\\_modelfit\(\)](#), [plot\\_pi\(\)](#), [plot\\_timetrend\(\)](#), [by\\_strata\\_DocTopic\(\)](#), [values\\_fig\(\)](#)

---

top_docs	<i>Show the top documents for each topic</i>
----------	--

---

**Description**

Show the top documents for each topic

**Usage**

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

**Arguments**

x	the output from a keyATM model (see <a href="#">keyATM()</a> ).
n	How many documents to show. Default is 10.

**Value**

An  $n \times k$  table of the top  $n$  documents for each topic, each number is a document index.

---

top_topics	<i>Show the top topics for each document</i>
------------	--

---

**Description**

Show the top topics for each document

**Usage**

```
top_topics(x, n = 2)
```

**Arguments**

x	the output from a keyATM model (see <a href="#">keyATM()</a> ).
n	integer. The number of topics to show. Default is 2.

**Value**

An  $n \times k$  table of the top  $n$  topics in each document.

---

top_words	<i>Show the top words for each topic</i>
-----------	--

---

### Description

If `show_keyword` is TRUE then words in their keyword topics are suffixed with a check mark. Words from another keyword topic are labeled with the name of that category.

### Usage

```
top_words(x, n = 10, measure = c("probability", "lift"), show_keyword = TRUE)
```

### Arguments

<code>x</code>	the output (see <a href="#">keyATM()</a> and <a href="#">by_strata_TopicWord()</a> ).
<code>n</code>	integer. The number terms to visualize. Default is 10.
<code>measure</code>	character. The way to sort the terms: <code>probability</code> (default) or <code>lift</code> .
<code>show_keyword</code>	logical. If TRUE, mark keywords. Default is TRUE.

### Value

An  $n \times k$  table of the top  $n$  words in each topic

---

values_fig	<i>Get values used to create a figure</i>
------------	---

---

### Description

Get values used to create a figure

### Usage

```
values_fig(x)
```

### Arguments

<code>x</code>	the <code>keyATM_fig</code> object.
----------------	-------------------------------------

### See Also

[save\\_fig\(\)](#), [visualize\\_keywords\(\)](#), [plot\\_alpha\(\)](#), [plot\\_modelfit\(\)](#), [plot\\_pi\(\)](#), [plot\\_timetrend\(\)](#), [by\\_strata\\_DocTopic\(\)](#)

---

visualize\_keywords      *Visualize keywords*

---

### Description

Visualize the proportion of keywords in the documents.

### Usage

```
visualize_keywords(docs, keywords, prune = TRUE, label_size = 3.2)
```

### Arguments

docs	a keyATM_docs object, generated by keyATM_read() function
keywords	a list of keywords
prune	logical. If TRUE, prune keywords that do not appear in docs. Default is TRUE.
label_size	the size of keyword labels in the output plot. Default is 3.2.

### Value

keyATM\_fig object

### See Also

[save\\_fig\(\)](#)

### Examples

```
## Not run:
# Prepare a keyATM_docs object
keyATM_docs <- keyATM_read(input)

# Keywords are in a list
keywords <- list(Education = c("education", "child", "student"),
                 Health     = c("public", "health", "program"))

# Visualize keywords
keyATM_viz <- visualize_keywords(keyATM_docs, keywords)

# View a figure
keyATM_viz

# Save a figure
save_fig(keyATM_viz, filename)

## End(Not run)
```

---

weightedLDA	<i>Weighted LDA main function</i>
-------------	-----------------------------------

---

## Description

Fit weighted LDA models.

## Usage

```
weightedLDA(
  docs,
  model,
  number_of_topics,
  model_settings = list(),
  priors = list(),
  options = list(),
  keep = c()
)
```

## Arguments

docs	texts read via <a href="#">keyATM_read()</a> .
model	Weighted LDA model: base, covariates, and dynamic.
number_of_topics	the number of regular topics.
model_settings	a list of model specific settings (details are in the online documentation).
priors	a list of priors of parameters.
options	a list of options (details are in the documentation of <a href="#">keyATM()</a> ).
keep	a vector of the names of elements you want to keep in output.

## Value

A keyATM\_output object containing:

**V** number of terms (number of unique words)

**N** number of documents

**model** the name of the model

**theta** topic proportions for each document (document-topic distribution)

**phi** topic specific word generation probabilities (topic-word distribution)

**topic\_counts** number of tokens assigned to each topic

**word\_counts** number of times each word type appears

**doc\_lens** length of each document in tokens

**vocab** words in the vocabulary (a vector of unique words)

**priors** priors  
**options** options  
**keywords\_raw** NULL for LDA models  
**model\_fit** perplexity and log-likelihood  
**pi** estimated pi for the last iteration (NULL for LDA models)  
**values\_iter** values stored during iterations  
**number\_of\_topics** number of topics  
**kept\_values** outputs you specified to store in keep option  
**information** information about the fitting

### See Also

`save.keyATM_output()`, [https://keyatm.github.io/keyATM/articles/pkgdown\\_files/Options.html](https://keyatm.github.io/keyATM/articles/pkgdown_files/Options.html)

### Examples

```
## Not run:
library(keyATM)
library(quanteda)
data(keyATM_data_bills)
bills_dfm <- keyATM_data_bills$doc_dfm # quanteda dfm object
keyATM_docs <- keyATM_read(bills_dfm)

# Weighted LDA
out <- weightedLDA(docs = keyATM_docs, model = "base",
                  number_of_topics = 5)

# Weighted LDA Covariates
bills_cov <- as.data.frame(keyATM_data_bills$cov)
out <- weightedLDA(docs = keyATM_docs, model = "covariates",
                  number_of_topics = 5,
                  model_settings = list(covariates_data = bills_cov,
                                       covariates_formula = ~ RepParty))

# Weighted LDA Dynamic
bills_time_index <- keyATM_data_bills$time_index
# Time index should start from 1 and increase by 1
bills_time_index <- as.integer(bills_time_index - 100)
out <- weightedLDA(docs = keyATM_docs, model = "dynamic",
                  number_of_topics = 5,
                  model_settings = list(num_states = 5,
                                       time_index = bills_time_index))

# Visit our website for full examples: https://keyatm.github.io/keyATM/

## End(Not run)
```

# Index

## \* datasets

- keyATM\_data\_bills, 9
- by\_strata\_DocTopic, 3
- by\_strata\_DocTopic(), 10, 11, 17, 19
- by\_strata\_TopicWord, 3
- by\_strata\_TopicWord(), 19
- covariates\_get, 4
- covariates\_info, 4
- dictionary, 16
- ggplot2::facet\_wrap(), 11, 14
- ggplot2::ggsave(), 17
- keyATM, 5
- keyATM(), 3, 4, 8, 11–19, 21
- keyATM-package, 2
- keyATM\_data\_bills, 9
- keyATM\_read, 9
- keyATM\_read(), 4, 5, 8, 16, 21
- keyATMvb, 7
- keyATMvb(), 17
- plot.strata\_doctopic, 10
- plot\_alpha, 11
- plot\_alpha(), 17, 19
- plot\_modelfit, 12
- plot\_modelfit(), 17, 19
- plot\_pi, 12
- plot\_pi(), 17, 19
- plot\_timetrend, 13
- plot\_timetrend(), 17, 19
- predict(), 3
- predict.keyATM\_output, 14
- quanteda::dictionary(), 16
- read\_keywords, 16
- save.keyATM\_output, 17
- save.keyATM\_output(), 7, 22
- save\_fig, 17
- save\_fig(), 11–14, 19, 20
- top\_docs, 18
- top\_topics, 18
- top\_words, 19
- values\_fig, 19
- values\_fig(), 17
- visualize\_keywords, 20
- visualize\_keywords(), 17, 19
- weightedLDA, 21
- weightedLDA(), 17