Package ‘keyATM’

June 11, 2022

Version 0.4.1

Title Keyword Assisted Topic Models

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 keyATM 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 (>= 1.0.7), dplyr (>= 1.0.0), fastmap, future.apply, ggplot2, ggrepel, magrittr, Matrix, matrixNormal (>= 0.1.0), MASS, pgdraw, 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 7.1.2

SystemRequirements C++11

NeedsCompilation yes

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

Maintainer Shusei Eshima <shuseieshima@g.harvard.edu>

Repository CRAN

Date/Publication 2022-06-11 15:40:05 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
multiPGreg ......................................................... 10
plot.strata_doctopic .......................................... 11
plot_alpha .......................................................... 12
plot_modelfit ...................................................... 13
plot_pi .............................................................. 13
plot_timetrend ..................................................... 14
predict.keyATM_output ....................................... 15
read_keywords .................................................... 16
save.keyATM_output ........................................... 17
save_fig ............................................................ 18
top_docs ............................................................ 18
top_topics .......................................................... 19
top_words ........................................................... 19
values_fig .......................................................... 20
visualize_keywords ............................................ 20
weightedLDA ......................................................... 21

Index 24

keyATM-package  Keyword Assisted Topic Models

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:

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

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 keyATM()).
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 predict.keyATM_output() 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)
covariates_info

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).

covariates_get  
Return covariates used in the iteration

description

Return covariates used in the iteration

Usage

covariates_get(x)

Arguments

- `x`: the output from the covariate keyATM model (see `keyATM()`)

covariates_info  
Show covariates information

description

Show covariates information

Usage

covariates_info(x)

Arguments

- `x`: the output from the covariate keyATM model (see `keyATM()`).
keyATM

keyATM main function

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 keyATM_read().
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
  • seed: A numeric value for random seed. If it is not provided, the package randomly selects a seed.
  • iterations: An integer. Number of iterations. Default is 1500.
  • verbose: If TRUE, it prints loglikelihood and perplexity. Default is FALSE.
  • llk_per: An integer. If the value is j keyATM stores loglikelihood and perplexity every j iteration. Default value is 10 per iterations
  • use_weights: If TRUE use weight. Default is TRUE.
  • weights_type: There are four types of weights. Weights based on the information theory (information-theory) and inverse frequency (inv-freq) and normalized versions of them (information-theory-normalized and inv-freq-normalized). Default is information-theory.
  • prune: If TRUE rume keywords that do not appear in the corpus. Default is TRUE.
- **store_theta**: If TRUE or 1, it stores $\theta$ (document-topic distribution) for the iteration specified by thinning. Default is FALSE (same as $\theta$).

- **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 $\theta$).

- **thinning**: An integer. If the value is $j$, keyATM stores following parameters every $j$ iteration. The default is 5.
  - $\theta$: For all models. If $\text{store}_\theta$ 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. Please p1an() before use this feature.

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
keyATMvb

See Also

save.keyATM_output(), https://keyatm.github.io/keyATM/articles/pkgdown_files/Options.html

Examples

```r
## 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)
```

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 keyATM_read()
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
    • convtol: the default is 1e-4
    • init: mcmc (default) or random
priors a list of priors of parameters
options a list of options same as keyATM(). 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
keyATM_data_bills

---

**keyATM_data_bills**  
**Bills data**

### 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](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](https://www.congress.gov)

---

**keyATM_read**  
**Read texts**

### Description

Read texts and create a keyATM_docs object, which is a list of texts.

### Usage

keyATM_read(
  texts,
  encoding = "UTF-8",
  check = TRUE,
  keep_docnames = FALSE,
  progress_bar = FALSE,
  split = 0
)

Arguments

texts input. keyATM takes a quanteda dfm (dgCMatrix), data.frame, tibble 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.
keep_docnames logical. If TRUE, it keeps the document names in a quanteda dfm. Default is FALSE.
progress_bar logical. If TRUE, it shows a progress bar (currently it only supports a quanteda object). Default is FALSE.
split numeric. This option works only with a quanteda dfm. It creates a two subset of the dfm by randomly splitting each document (i.e., the total number of documents is the same between two subsets). This option specifies the split proportion. Default is 0.

Value

a keyATM_docs object. The first element is a list whose elements are split 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)

multiPReg Run multinomial regression with Polya-Gamma augmentation

Description

Run multinomial regression with Polya-Gamma augmentation. There is no need to call this function directly. The keyATM Covariate internally uses this.
Usage

multiPGreg(Y, X, num_topics, PG_params, iter = 1, store_lambda = 0)

Arguments

Y  Outcomes.
X  Covariates.
num_topics  Number of topics.
PG_params  Parameters used in this function.
iter  The default is 1.
store_lambda  The default is 0.

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_modelfit

Show a diagnosis plot of log-likelihood and perplexity

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

Show a diagnosis plot of pi

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 `keyATM()`).
- **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 `keyATM()`).
- **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 `keyATM()`).
- **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 `keyATM()`).

Value

- **keyATM_fig** object.

See Also

- `save_fig()`

### plot_timetrend

**Plot time trend**

### Description

Plot time trend

### Usage

```r
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",
  show_point = TRUE,
  ...
)
```
predict.keyATM_output

Arguments

- **x**: the output from the dynamic keyATM model (see `keyATM()`).
- **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 `keyATM()`).
- **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 `keyATM()`).
- **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 `keyATM()`).
- **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 `keyATM()`).
- **xlab**: a character.
- **scales**: character. Control the scale of y-axis (the parameter in `ggplot2::facet_wrap()`): free adjusts y-axis for parameters. Default is fixed.
- **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()`

Description

Predict topic proportions for the covariate keyATM

Usage

```r
## S3 method for class 'keyATM_output'
predict(
  object,
  newdata,
  transform = FALSE,
  ...)
arguments

object the keyATM_output object for the covariate model.
newdata New observations which should be predicted.
transform Transform and standardize the newdata with the same formula and option as model_settings used in keyATM().
burn_in integer. Burn-in period. If not specified, it is the half of samples. Default is NULL.
parallel logical. If TRUE, parallelization for speeding up. Default is TRUE. Please plan() before use this function.
posterior_mean logical. If TRUE, the quantity of interest to estimate is the posterior mean. Default is TRUE.
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.
label a character. Add a label column to the output. The default is NULL (do not add it).
raw_values a logical. Returns raw values. The default is FALSE.
... additional arguments not used.

read_keywords Convert a quanteda dictionary to keywords

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, ...)
save.keyATM_output

Arguments

file  file identifier for a foreign dictionary, e.g. path to a dictionary in YAML or LIWC format

docs  texts read via keyATM_read()

dictionary  a quanteda dictionary object, ignore if file is not NULL

split  boolean, if multi-word terms be seperated, e.g. "air force" splits into "air" and "force".

...  additional parameters for quanteda::dictionary()

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 dictionary example from quanteda
dictfile <- tempfile()
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)

Description

Save a keyATM_output object

Usage

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

Arguments

x  a keyATM_output object (see keyATM()).

file  file name to create on disk.
See Also

keyATM(), weightedLDA(), keyATMvb()

save_fig

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 ggplot2::ggsave() function.

See Also

visualize_keywords(), plot_alpha(), plot_modelfit(), plot_pi(), plot_timetrend(), by_strata_DocTopic(), values_fig()

top_docs

Description

Show the top documents for each topic

Usage

top_docs(x, n = 10)

Arguments

x the output from a keyATM model (see keyATM()).
n How many documents to show. Default is 10.

Value

An n x k table of the top n documents for each topic, each number is a document index.
### top_topics

**Show the top topics for each document**

**Description**

Show the top topics for each document

**Usage**

top_topics(x, n = 2)

**Arguments**

- `x`: the output from a keyATM model (see `keyATM()`).
- `n`: integer. The number of topics to show. Default is 2.

**Value**

An n x k table of the top n topics in each document.

### top_words

**Show the top words for each topic**

**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**

- `x`: the output (see `keyATM()` and `by_strata_TopicWord()`).
- `n`: integer. The number terms to visualize. Default is 10.
- `measure`: character. The way to sort the terms: probability (default) or lift.
- `show_keyword`: logical. If TRUE, mark keywords. Default is TRUE.

**Value**

An n x k table of the top n words in each topic
values_fig  Get values used to create a figure

Description
Get values used to create a figure

Usage
values_fig(x)

Arguments
x  the keyATM_fig 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

```r
## 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

Weighted LDA main function

Description

Fit weighted LDA models.

Usage

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

Arguments

docs texts read via `keyATM_read()`.
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.
weightedLDA

options a list of options (details are in the documentation of `keyATM()`).
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

Examples

```r
## Not run:
library(keyATM)
library(quanteda)
data(keyATM_data_bills)
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(), 11, 12, 18, 20
by_strata_TopicWord, 3
by_strata_TopicWord(), 19

covariates_get, 4
covariates_info, 4
dictionary, 17
ggplot2::facet_wrap(), 12, 15
ggplot2::ggsave(), 18

keyATM, 5
keyATM(), 3, 4, 8, 12–19, 22
keyATM-package, 2
keyATM_data_bills, 9
keyATM_read, 9
keyATM_read(), 4, 5, 8, 17, 21
keyATMvb, 7
keyATMvb(), 18

multiPGreg, 10

plot.strata_doctopic, 11
plot_alpha, 12
plot_alpha(), 18, 20
plot_modelfit, 13
plot_modelfit(), 18, 20
plot_pi, 13
plot_pi(), 18, 20
plot_timetrend, 14
plot_timetrend(), 18, 20
predict.keyATM_output, 15
predict.keyATM_output(), 3

quanteda::dictionary(), 17

read_keywords, 16

save.keyATM_output, 17
save.keyATM_output(), 7, 22
save_fig, 18
save_fig(), 12–15, 20
top_docs, 18
top_topics, 19
top_words, 19
values_fig, 20
values_fig(), 18
visualize_keywords, 20
visualize_keywords(), 18, 20

weightedLDA, 21
weightedLDA(), 18

24