

Package ‘BERTopic’

November 1, 2025

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License MIT + file LICENSE

Encoding UTF-8

Imports reticulate,
rlang,
tibble,

Suggests Matrix,
htmltools,
testthat (>= 3.1.0),

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 3.5)

LazyData true

Contents

as.data.frame.bertopic_r	2
bertopic_as_document_topic_matrix	3
bertopic_available	3
bertopic_find_topics	4
bertopic_fit	4
bertopic_get_document_info	5
bertopic_get_representative_docs	5
bertopic_has_embedding_model	6
bertopic_load	6
bertopic_reduce_topics	7
bertopic_save	7
bertopic_self_check	8
bertopic_session_info	9
bertopic_set_embedding_model	9
bertopic_set_topic_labels	10
bertopic_topics	10
bertopic_topics_over_time	11
bertopic_topic_terms	11

bertopic_transform	12
bertopic_update_topics	12
bertopic_visualize_barchart	13
bertopic_visualize_documents	13
bertopic_visualize_heatmap	14
bertopic_visualize_hierarchy	14
bertopic_visualize_term_rank	14
bertopic_visualize_topics	15
bertopic_visualize_topics_over_time	15
coef.bertopic_r	16
fortify.bertopic_r	16
install_py_deps	17
install_py_deps_conda	17
install_py_deps_venv	18
predict.bertopic_r	19
print.bertopic_r	19
set_bertopic_seed	20
sms_spam	20
summary.bertopic_r	21
use_bertopic	21
use_bertopic_condaenv	22
use_bertopic_virtualenv	22

Index**24****as.data.frame.bertopic_r***Coerce to data.frame***Description**

Coerce to data.frame

Usage

```
## S3 method for class 'bertopic_r'
as.data.frame(x, ...)
```

Arguments

- x A "bertopic_r" model.
- ... Unused.

ValueA data.frame equal to `bertopic_topics()`.

bertopic_as_document_topic_matrix
Coerce to a document-topic probability matrix

Description

Extract the document-topic probabilities as a matrix. If probabilities were not computed during fitting, returns NULL (with a warning).

Usage

```
bertopic_as_document_topic_matrix(model, sparse = TRUE, prefix = TRUE)
```

Arguments

model	A "bertopic_r" model.
sparse	Logical; if TRUE and Matrix is available, returns a sparse matrix.
prefix	Logical; if TRUE, prefix columns as "topic_".

Value

A matrix or sparse Matrix of size n_docs x n_topics, or NULL.

bertopic_available *Is Python + BERTopic available?*

Description

Checks whether the active Python (as initialized by **reticulate**) can import the key modules needed for BERTopic.

Usage

```
bertopic_available()
```

Value

Logical scalar.

Examples

```
## Not run:  
bertopic_available()  
  
## End(Not run)
```

`bertopic_find_topics` *Find nearest topics for a query string*

Description

Use `BERTopic.find_topics()` to retrieve the closest topics for a query string. Augments topic IDs/scores with topic labels when available.

Usage

```
bertopic_find_topics(model, query_text, top_n = 5L)
```

Arguments

<code>model</code>	A "bertopic_r" model.
<code>query_text</code>	A length-1 character query.
<code>top_n</code>	Number of nearest topics to return.

Value

A tibble with columns `topic`, `score`, and `label`.

`bertopic_fit` *Fit BERTopic from R*

Description

A high-level wrapper around Python 'BERTopic'. Python dependencies are checked at runtime.

Usage

```
bertopic_fit(text, embeddings = NULL, ...)
```

Arguments

<code>text</code>	Character vector of documents.
<code>embeddings</code>	Optional numeric matrix (n_docs x dim). If supplied, passed through to Python.
<code>...</code>	Additional arguments forwarded to <code>bertopic.BERTopic(...)</code> .

Value

An S3 object of class "bertopic_r" containing:

- `.py`: the underlying Python model (reticulate object)
- `topics`: integer vector of topic assignments
- `probs`: numeric matrix/data frame of topic probabilities (if available)

Examples

```
## Not run:
if (reticulate::py_module_available("bertopic")) {
  m <- bertopic_fit(c("a doc", "another doc"))
  print(class(m))
}

## End(Not run)
```

bertopic_get_document_info

Document-level information

Description

Retrieve document-level information for the provided documents.

Usage

```
bertopic_get_document_info(model, docs)
```

Arguments

model	A "bertopic_r" model.
docs	Character vector of documents to query (required).

Value

A tibble with document-level information.

bertopic_get_representative_docs

Representative documents for a topic

Description

Retrieve representative documents for a given topic using `BERTopic.get_representative_docs()`. Falls back across signature variants.

Usage

```
bertopic_get_representative_docs(model, topic_id, top_n = 5L)
```

Arguments

model	A "bertopic_r" model.
topic_id	Integer topic id.
top_n	Number of representative documents to return.

Value

A tibble with columns rank and document. If scores are available in the current BERTopic version, a score column is included.

`bertopic_has_embedding_model`

Does the model have a usable embedding model?

Description

Does the model have a usable embedding model?

Usage

`bertopic_has_embedding_model(model)`

Arguments

`model` A "bertopic_r" model.

Value

Logical; TRUE if `embedding_model` is present and not None.

`bertopic_load`

Load a BERTopic model

Description

Load a BERTopic model from disk that was saved with `bertopic_save()`.

Usage

`bertopic_load(path)`

Arguments

`path` Path used in `bertopic_save()` (file or directory).

Value

A "bertopic_r" object with the loaded Python model.

<code>bertopic_reduce_topics</code>	<i>Reduce/merge topics</i>
-------------------------------------	----------------------------

Description

Wrapper over Python `reduce_topics`, compatible with multiple signatures.

Usage

```
bertopic_reduce_topics(
    model,
    nr_topics = "auto",
    representation_model = NULL,
    docs = NULL
)
```

Arguments

model	A "bertopic_r" model.
nr_topics	Target number (integer) or "auto".
representation_model	Optional Python representation model.
docs	Optional character vector of training docs (used if required by backend).

Value

The input model (invisibly).

<code>bertopic_save</code>	<i>Save a BERTTopic model</i>
----------------------------	-------------------------------

Description

Save a fitted BERTTopic model to disk. Depending on the serialization method, this may produce either a single file (e.g., `*.pkl` / `*.pt` / `*.safetensors`) or a directory bundle. The function does not pre-create the target path; it only ensures the parent directory exists and lets BERTTopic decide the layout.

Usage

```
bertopic_save(
    model,
    path,
    serialization = c("pickle", "safetensors", "pt"),
    save_embedding_model = FALSE,
    overwrite = FALSE
)
```

Arguments

model	A "bertopic_r" model.
path	Destination path (file or directory, as required by BERTopic).
serialization	One of "pickle", "safetensors", or "pt". Default "pickle".
save_embedding_model	Logical; whether to include the embedding model. Default FALSE.
overwrite	Logical; if TRUE and the target exists, it will be replaced.

Value

Invisibly returns the normalized path.

bertopic_self_check *Quick self-check for the BERTopic R interface*

Description

Runs a quick end-to-end smoke test:

- Report Python path/version.
- Verify that `bertopic` is importable and report its version.
- Minimal round trip: `fit` -> `transform` -> `save` -> `load`.

Usage

```
bertopic_self_check()
```

Value

A named list with fields:

python_ok Logical.
bertopic_ok Logical.
roundtrip_ok Logical.
details Character vector of diagnostic messages.

Examples

```
## Not run:  
bertopic_self_check()  
  
## End(Not run)
```

bertopic_session_info *Summarize Python/BERTopic session info*

Description

Summarize Python/BERTopic session info

Usage

```
bertopic_session_info()
```

Value

A named list containing paths, versions, and module availability:

python Path of the active Python.
libpython Path to libpython, if any.
version Python version string.
numpy Whether NumPy is available.
numpy_version NumPy version string (if available).
modules A data.frame with availability for key modules.

Examples

```
## Not run:  
bertopic_session_info()  
  
## End(Not run)
```

bertopic_set_embedding_model
Replace or set the embedding model

Description

Set a new embedding model on a fitted BERTTopic instance. This enables `transform()` after loading when the embedding model was not saved.

Usage

```
bertopic_set_embedding_model(model, embedding_model)
```

Arguments

model	A "bertopic_r" model.
embedding_model	Either a character identifier (e.g., "all-MiniLM-L6-v2") or a Python embedding model object (e.g., a SentenceTransformer instance).

Value

The input model (invisibly).

`bertopic_set_topic_labels`
Relabel topics

Description

Set custom labels for topics. Accepts a named character vector or a data.frame with columns `topic` and `label`.

Usage

```
bertopic_set_topic_labels(model, labels)
```

Arguments

<code>model</code>	A "bertopic_r" model.
<code>labels</code>	A named character vector (names are topic ids) or a data.frame.

Value

The input model (invisibly).

`bertopic_topics` *Get topic info as a tibble*

Description

Get topic info as a tibble

Usage

```
bertopic_topics(model)
```

Arguments

<code>model</code>	A "bertopic_r" object returned by bertopic_fit() .
--------------------	--

Value

A tibble with topic-level information from Python `get_topic_info()`.

`bertopic_topics_over_time`
Compute topics over time

Description

Wrapper for Python BERTopic.`topics_over_time()`. Returns a tibble and attaches the original Python dataframe in the "`_py`" attribute for use in visualization.

Usage

```
bertopic_topics_over_time(  
  model,  
  docs,  
  timestamps,  
  nr_bins = NULL,  
  datetime_format = NULL  
)
```

Arguments

<code>model</code>	A "bertopic_r" model.
<code>docs</code>	Character vector of documents.
<code>timestamps</code>	A vector of timestamps (Date, POSIXt, or character).
<code>nr_bins</code>	Optional number of temporal bins.
<code>datetime_format</code>	Optional strftime-style format if timestamps are strings.

Value

A tibble with topics-over-time data; attribute "`_py`" stores the original Python dataframe.

`bertopic_topic_terms` *Get top terms for a topic*

Description

Get top terms for a topic

Usage

```
bertopic_topic_terms(model, topic_id, top_n = 10L)
```

Arguments

<code>model</code>	A "bertopic_r" model
<code>topic_id</code>	Integer topic id
<code>top_n</code>	Number of top terms to return

Value

A tibble with columns `term` and `weight`

<code>bertopic_transform</code>	<i>Transform new documents with a fitted BERTopic model</i>
---------------------------------	---

Description

Transform new documents with a fitted BERTopic model

Usage

```
bertopic_transform(model, new_text, embeddings = NULL)
```

Arguments

- `model` A "bertopic_r" model from [bertopic_fit\(\)](#).
- `new_text` Character vector of new documents.
- `embeddings` Optional numeric matrix for new documents.

Value

A list with `topics` and `probs` for the new documents.

<code>bertopic_update_topics</code>	<i>Update topic representations</i>
-------------------------------------	-------------------------------------

Description

Call Python `BERTopic.update_topics()` to recompute topic representations.

Usage

```
bertopic_update_topics(model, text)
```

Arguments

- `model` A "bertopic_r" model.
- `text` Character vector of training documents used in `fit`.

Value

The input model (invisibly), updated in place on the Python side.

bertopic_visualize_barchart
Visualize a topic barchart

Description

Visualize a topic barchart

Usage

```
bertopic_visualize_barchart(model, topic_id = NULL, file = NULL)
```

Arguments

model	A "bertopic_r" model.
topic_id	Integer topic id. If NULL, a set of top topics is shown.
file	Optional HTML output path.

bertopic_visualize_documents
Visualize embedded documents

Description

Visualize embedded documents

Usage

```
bertopic_visualize_documents(model, docs = NULL, file = NULL)
```

Arguments

model	A "bertopic_r" model.
docs	Optional character vector of documents to visualize.
file	Optional HTML output path.

bertopic_visualize_heatmap
Visualize topic similarity heatmap

Description

Visualize topic similarity heatmap

Usage

```
bertopic_visualize_heatmap(model, file = NULL)
```

Arguments

model	A "bertopic_r" model.
file	Optional HTML output path.

bertopic_visualize_hierarchy
Visualize hierarchical clustering of topics

Description

Visualize hierarchical clustering of topics

Usage

```
bertopic_visualize_hierarchy(model, file = NULL)
```

Arguments

model	A "bertopic_r" model.
file	Optional HTML output path.

bertopic_visualize_term_rank
Visualize term rank evolution

Description

Visualize term rank evolution

Usage

```
bertopic_visualize_term_rank(model, file = NULL)
```

Arguments

model	A "bertopic_r" model.
file	Optional HTML output path.

bertopic_visualize_topics
Visualize topic map

Description

Visualize topic map

Usage

```
bertopic_visualize_topics(model, file = NULL)
```

Arguments

model	A "bertopic_r" model.
file	Optional HTML output path. If NULL, returns htmltools::HTML.

bertopic_visualize_topics_over_time
Visualize topics over time

Description

Visualize topics over time

Usage

```
bertopic_visualize_topics_over_time(  
  model,  
  topics_over_time,  
  top_n = 10L,  
  file = NULL  
)
```

Arguments

model	A "bertopic_r" model.
topics_over_time	A tibble returned by bertopic_topics_over_time() , or a Python dataframe compatible with <code>visualize_topics_over_time()</code> .
top_n	Number of topics to display.
file	Optional HTML output path.

`coef.bertopic_r` *Coefficients (top terms) for BERTopic*

Description

Coefficients (top terms) for BERTTopic

Usage

```
## S3 method for class 'bertopic_r'
coef(object, top_n = 10L, ...)
```

Arguments

<code>object</code>	A "bertopic_r" model.
<code>top_n</code>	Number of terms per topic.
<code>...</code>	Unused.

Value

A data.frame with columns topic, term, weight.

`fortify.bertopic_r` *Fortify method for ggplot2*

Description

Fortify method for ggplot2

Usage

```
fortify.bertopic_r(model, data, ...)
```

Arguments

<code>model</code>	A "bertopic_r" model.
<code>data</code>	Ignored.
<code>...</code>	Unused.

Value

A data.frame of document-topic assignments.

<code>install_py_deps</code>	<i>Install Python dependencies for BERTopic (auto route)</i>
------------------------------	--

Description

Tries Conda first (recommended). If Conda is unavailable, falls back to virtualenv. On success, prints which route was used.

Usage

```
install_py_deps(
  envname = "r-bertopic",
  python_version = "3.10",
  python = NULL,
  reinstall = FALSE,
  validate = TRUE,
  verbose = TRUE
)
```

Arguments

<code>envname</code>	Character. Environment name (both routes). Default "r-bertopic".
<code>python_version</code>	Character. Python version for Conda route, e.g. "3.10".
<code>python</code>	Optional path to python for virtualenv route.
<code>reinstall</code>	Logical. Recreate the environment if it exists (route-specific).
<code>validate</code>	Logical. Attempt to validate imports if reticulate is not already initialized to another Python.
<code>verbose</code>	Logical. Print progress.

Value

Invisibly, the path to the selected Python interpreter.

<code>install_py_deps_conda</code>	<i>Install Python dependencies for BERTopic (Conda route)</i>
------------------------------------	---

Description

Creates (or reuses) a Conda environment with a pinned Python toolchain, installs the scientific stack + PyTorch (CPU) + sentence-transformers, then installs `bertopic==0.16.0` via pip. Optionally validates imports.

Usage

```
install_py_deps_conda(
  envname = "r-bertopic",
  python_version = "3.10",
  reinstall = FALSE,
  validate = TRUE,
  verbose = TRUE
)
```

Arguments

<code>envname</code>	Character. Conda environment name. Default "r-bertopic".
<code>python_version</code>	Character. Python version to use, e.g. "3.10".
<code>reinstall</code>	Logical. If TRUE, delete any existing env and recreate.
<code>validate</code>	Logical. If TRUE, bind and validate imports (will skip if reticulate is already initialized to another Python).
<code>verbose</code>	Logical. Print progress messages.

Value

Invisibly returns the path to the Python executable inside the env.

Examples

```
## Not run:
install_py_deps_conda(envname = "r-bertopic", python_version = "3.10")

## End(Not run)
```

`install_py_deps_venv` *Install Python dependencies for BERTopic (virtualenv route)*

Description

Creates (or reuses) a virtualenv and installs bertopic==0.16.0 plus required dependencies via pip. Optionally validates imports.

Usage

```
install_py_deps_venv(
  envname = "r-bertopic",
  python = NULL,
  reinstall = FALSE,
  validate = TRUE,
  verbose = TRUE
)
```

Arguments

<code>envname</code>	Character. Virtualenv name. Default "r-bertopic".
<code>python</code>	Character. Path to a Python executable to create the venv with. If NULL, tries to find python / python3 on PATH.
<code>reinstall</code>	Logical. If TRUE, delete existing venv and recreate.
<code>validate</code>	Logical. If TRUE, bind and validate imports (will skip if reticulate is already initialized to another Python).
<code>verbose</code>	Logical. Print progress messages.

Value

Invisibly returns the path to the Python executable inside the venv.

Examples

```
## Not run:
install_py_deps_venv(envname = "r-bertopic")

## End(Not run)
```

predict.bertopic_r *Predict method for BERTTopic models*

Description

Predict method for BERTTopic models

Usage

```
## S3 method for class 'bertopic_r'
predict(
  object,
  newdata,
  type = c("both", "class", "prob"),
  embeddings = NULL,
  ...
)
```

Arguments

object	A "bertopic_r" model.
newdata	Character vector of new documents.
type	One of "class", "prob", or "both".
embeddings	Optional numeric matrix of embeddings.
...	Reserved for future arguments.

Value

Depending on type, an integer vector, a matrix/data frame, or a list.

print.bertopic_r *Print method for bertopic_r*

Description

Print method for bertopic_r

Usage

```
## S3 method for class 'bertopic_r'
print(x, ...)
```

Arguments

- x A "bertopic_r" object.
- ... Unused.

set_bertopic_seed *Set random seed for R and Python backends*

Description

Set random seed for R and Python backends

Usage

```
set_bertopic_seed(seed)
```

Arguments

- seed Integer seed

sms_spam *SMS Spam Collection (UCI) - subset for examples*

Description

A cleaned subset of the UCI SMS Spam Collection, suitable for quick examples and tests in this package. Each row is an SMS message labeled as "ham" or "spam".

Usage

```
sms_spam
```

Format

A data frame with two columns:

label Character, either "ham" or "spam".

text Character, the SMS message content (UTF-8).

Note

This dataset is included for educational/demo purposes. If you use it in publications, please cite the original authors and the UCI repository page.

Source

UCI Machine Learning Repository: SMS Spam Collection. Dataset page: <https://archive.ics.uci.edu/dataset/228/sms+spam+collection> Original citation: Almeida, T.A., Hidalgo, J.M.G., & Yamakami, A. (2011). Contributions to the Study of SMS Spam Filtering: New Collection and Results.

Examples

```
data(sms_spam)
head(sms_spam)
```

summary.bertopic_r *Summary for BERTopic models*

Description

Summary for BERTopic models

Usage

```
## S3 method for class 'bertopic_r'
summary(object, ...)
```

Arguments

object	A "bertopic_r" model.
...	Unused.

Value

Invisibly returns a named list of summary fields.

use_bertopic *Bind current R session to the BERTopic environment (auto route)*

Description

If a Conda env with the given name exists, prefer Conda; otherwise try a virtualenv with the same name. Stops if neither exists.

Usage

```
use_bertopic(envname = "r-bertopic")
```

Arguments

envname	Character. Environment name. Default "r-bertopic".
---------	--

Value

Invisibly, the Python executable path.

`use_bertopic_condaenv` *Bind current R session to a BERTopic Conda environment*

Description

Sets RETICULATE_PYTHON to the environment's Python and initializes **reticulate**. If **reticulate** is already initialized to a different Python, this stops with an informative error.

Usage

```
use_bertopic_condaenv(envname = "r-bertopic", required = TRUE)
```

Arguments

<code>envname</code>	Character. Conda env name (default "r-bertopic").
<code>required</code>	Logical. Kept for API symmetry; unused.

Value

Invisibly returns the Python executable path in the env.

Examples

```
## Not run:
use_bertopic_condaenv("r-bertopic")

## End(Not run)
```

`use_bertopic_virtualenv`

Bind current R session to a BERTopic virtualenv

Description

Sets RETICULATE_PYTHON to the Python inside the given virtualenv and initializes **reticulate**. If **reticulate** is already initialized to a different Python, this stops with an informative error.

Usage

```
use_bertopic_virtualenv(envname = "r-bertopic", required = TRUE)
```

Arguments

<code>envname</code>	Character. Virtualenv name (default "r-bertopic").
<code>required</code>	Logical. Kept for API symmetry; unused.

Value

Invisibly returns the Python executable path in the env.

Examples

```
## Not run:  
use_bertopic_virtualenv("r-bertopic")  
  
## End(Not run)
```

Index

- * **datasets**
 - sms_spam, 20
- as.data.frame.bertopic_r, 2
- bertopic_as_document_topic_matrix, 3
- bertopic_available, 3
- bertopic_find_topics, 4
- bertopic_fit, 4
- bertopic_fit(), 10, 12
- bertopic_get_document_info, 5
- bertopic_get_representative_docs, 5
- bertopic_has_embedding_model, 6
- bertopic_load, 6
- bertopic_reduce_topics, 7
- bertopic_save, 7
- bertopic_save(), 6
- bertopic_self_check, 8
- bertopic_session_info, 9
- bertopic_set_embedding_model, 9
- bertopic_set_topic_labels, 10
- bertopic_topic_terms, 11
- bertopic_topics, 10
- bertopic_topics(), 2
- bertopic_topics_over_time, 11
- bertopic_topics_over_time(), 15
- bertopic_transform, 12
- bertopic_update_topics, 12
- bertopic_visualize_barchart, 13
- bertopic_visualize_documents, 13
- bertopic_visualize_heatmap, 14
- bertopic_visualize_hierarchy, 14
- bertopic_visualize_term_rank, 14
- bertopic_visualize_topics, 15
- bertopic_visualize_topics_over_time,
 15
- coef.bertopic_r, 16
- fortify.bertopic_r, 16
- install_py_deps, 17
- install_py_deps_conda, 17
- install_py_deps_venv, 18
- predict.bertopic_r, 19
- print.bertopic_r, 19
- set_bertopic_seed, 20
- sms_spam, 20
- summary.bertopic_r, 21
- use_bertopic, 21
- use_bertopic_condaenv, 22
- use_bertopic_virtualenv, 22