

Package ‘LDAShiny’

June 8, 2026

Title Interactive Topic Modeling and Bibliometric Analysis via Shiny

Version 1.0.0

Description Provides a 'Shiny' graphical interface for the complete workflow of Latent Dirichlet Allocation (LDA) topic modelling on bibliometric data from Scopus and Web of Science. Steps include data import and deduplication, text preprocessing (stopword removal, stemming, n-grams, sparse-term filtering), statistical inference to select the optimal number of topics via coherence, final model training, and topic trend analysis over time using linear regression. All results can be exported as Excel files, RDS objects, and publication-quality plots.

License GPL-3

URL <https://github.com/JavierDeLaHoz/LDAShiny>

Depends R (>= 4.1.0)

Imports colourpicker, config (>= 0.3.1), dplyr, DT, ggplot2, golem (>= 0.4.0), Matrix, openxlsx, quanteda, RColorBrewer, readxl, shiny (>= 1.7.0), shinybusy, shinydashboard, shinyjs, shinyWidgets, slam, SnowballC, stopwords, textmineR, tibble, tidyr, tm, wordcloud, broom, parallel, stats, utils, grDevices

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Encoding UTF-8

RoxygenNote 7.3.3

Config/testthat/edition 3

Language en-US

VignetteBuilder knitr

NeedsCompilation no

Author Javier De La Hoz-M [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-7779-0803>>)

Maintainer Javier De La Hoz-M <jdelahoz@unimagdalena.edu.co>

Repository CRAN

Date/Publication 2026-06-08 06:50:14 UTC

Contents

mod_about_server	2
mod_about_ui	2
mod_import_bibliometrics_server	3
mod_import_bibliometrics_ui	3
mod_inference_server	4
mod_inference_ui	4
mod_lda_train	5
mod_preprocess_server	5
mod_preprocess_ui	6
mod_trend_server	6
mod_trend_ui	7
run_LDAShiny	7

Index	9
--------------	----------

mod_about_server	<i>About Server</i>
------------------	---------------------

Description

About Server

Usage

mod_about_server(id)

Arguments

id	Shiny module id
----	-----------------

mod_about_ui	<i>About UI</i>
--------------	-----------------

Description

About UI

Usage

mod_about_ui(id)

Arguments

id	Shiny module id
----	-----------------

mod_import_bibliometrics_server
Import Bibliometrics Server

Description

Server logic for importing, validating and combining Scopus/WoS datasets.

Usage

```
mod_import_bibliometrics_server(id)
```

Arguments

id Shiny module ID.

Value

A list with one element:

data A reactive expression returning the cleaned combined data.frame.

mod_import_bibliometrics_ui
Import Bibliometrics UI

Description

Shiny module UI for importing and validating bibliometric datasets from Scopus and Web of Science in LDAShiny.

Usage

```
mod_import_bibliometrics_ui(id)
```

Arguments

id Shiny module ID.

Value

A Shiny UI element (tagList).

mod_inference_server *LDA Inference Module Server*

Description

LDA Inference Module Server

Usage

```
mod_inference_server(id, dtm_reactive)
```

Arguments

`id` Shiny module ID.
`dtm_reactive` A reactive expression returning a dgCMatrix DTM.

Value

A list with:

model_list Reactive list of fitted LDA models, one per k.
coherence_mat Reactive data.frame with columns k and coherence.
k_optimal Reactive integer: k with highest mean coherence.

mod_inference_ui *LDA Inference Module UI*

Description

UI for running multiple LDA models over a range of topic counts (k) and selecting the best k via mean coherence.

Usage

```
mod_inference_ui(id)
```

Arguments

`id` Shiny module ID.

Value

A Shiny UI element (tagList).

mod_lda_train	<i>Final LDA Model Training Module</i>
---------------	--

Description

Trains the final LDA model using the processed DTM. The number of topics (K) is automatically initialised from the inference module but can be overridden.

Usage

```
mod_lda_train()
```

```
mod_lda_train_ui(id)
```

```
mod_lda_train_server(id, dtm_reactive, k_optimal_reactive, bd_reactive)
```

Arguments

id Shiny module ID.

dtm_reactive Reactive returning the DTM (dgCMatrix).

k_optimal_reactive

Reactive returning the optimal K (integer).

bd_reactive Reactive returning the original metadata data.frame.

Value

A list of reactive values with the LDA model outputs.

mod_preprocess_server	<i>Text Preprocessing Module Server</i>
-----------------------	---

Description

Text Preprocessing Module Server

Usage

```
mod_preprocess_server(id, data_reactive)
```

Arguments

id Shiny module ID.

data_reactive A reactive expression returning the input data.frame.

Value

A list with:

dtm Reactive returning the filtered sparse DTM (`dgCMatrix`).

tf_mat Reactive returning the term-frequency data.frame.

mod_preprocess_ui *Text Preprocessing Module UI*

Description

Provides UI controls to clean, tokenize, stem, remove stopwords, build a Document-Term Matrix (DTM), and filter sparse terms.

Usage

```
mod_preprocess_ui(id)
```

Arguments

`id` Shiny module ID.

Value

A Shiny UI element (`tagList`).

mod_trend_server *Topic Trend Analysis Module Server*

Description

Topic Trend Analysis Module Server

Usage

```
mod_trend_server(id, df_year_topic_reactive, lda_model_reactive)
```

Arguments

`id` Shiny module ID.

`df_year_topic_reactive`

Reactive returning a data.frame with columns Topic, year, and Intensity.

`lda_model_reactive`

Reactive returning the final LDA model object (used for completeness; labels are derived from data columns only).

Value

NULL invisibly.

mod_trend_ui	<i>Topic Trend Analysis Module UI</i>
--------------	---------------------------------------

Description

Performs linear regression on topic intensity (mean Gamma) over years and classifies topics as increasing (HOT), decreasing (COLD), or stable (EQUAL).

Usage

```
mod_trend_ui(id)
```

Arguments

id Shiny module ID.

Value

A Shiny UI element (tagList).

run_LDAShiny	<i>Run the LDAShiny Application</i>
--------------	-------------------------------------

Description

Launches the LDAShiny Shiny application using the golem framework. The function temporarily raises the maximum upload size to 500 MB for the duration of the session and restores the previous value on exit.

Usage

```
run_LDAShiny(max_upload_mb = 500, ...)
```

Arguments

max_upload_mb Maximum file upload size in megabytes (default 500). Passed to options(shiny.maxRequestSize = ...).

... Additional named options forwarded to [with_golem_options](#).

Value

A Shiny app object (invisibly).

Examples

```
if (interactive()) {  
  run_LDAShiny()  
}
```

Index

`mod_about_server`, [2](#)
`mod_about_ui`, [2](#)
`mod_import_bibliometrics_server`, [3](#)
`mod_import_bibliometrics_ui`, [3](#)
`mod_inference_server`, [4](#)
`mod_inference_ui`, [4](#)
`mod_lda_train`, [5](#)
`mod_lda_train_server` (`mod_lda_train`), [5](#)
`mod_lda_train_ui` (`mod_lda_train`), [5](#)
`mod_preprocess_server`, [5](#)
`mod_preprocess_ui`, [6](#)
`mod_trend_server`, [6](#)
`mod_trend_ui`, [7](#)

`run_LDAShiny`, [7](#)

`with_golem_options`, [7](#)