Package: LDAShiny (via r-universe)

October 30, 2026

Title Shiny Application for Topic Modeling of Scientific Literature

Version 1.1.0

Description A 'shiny' application that guides users through an exploratory review of scientific literature using Latent Dirichlet Allocation (LDA) topic modeling. It imports bibliometric records exported from databases such as 'Scopus' and 'Web of Science', performs text preprocessing, helps select the number of topics, fits the final model, and visualizes the results and the evolution of topics over time. The methodology is described in De la Hoz-M et al. (2021) <doi:10.3390/math9141671>.

License GPL-3

URL https://github.com/jdelahoz-m/LDAShiny

BugReports https://github.com/jdelahoz-m/LDAShiny/issues

Depends R (>= 4.1.0)

Imports beepr, broom, colourpicker, config, dplyr, DT, ggplot2, ggrepel, golem, igraph, jsonlite, LDAvis, Matrix, openxlsx, quanteda, RColorBrewer, readxl, scales, shiny, shinybusy, shinydashboard, shinyjs, shinyWidgets, slam, SnowballC, stopwords, textmineR, tibble, tidyr, tm, utils, wordcloud

Suggests knitr, plotly, purrr, readr, rmarkdown, shinyBS, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Javier De La Hoz-M [aut, cre] (ORCID: <https://orcid.org/0000-0001-7779-0803>), Maria Camila Samper Meza [aut] (ORCID: <https://orcid.org/0009-0005-9235-8732>),

Alexander Armando Bustamante Martínez [aut] (ORCID:
<https://orcid.org/0000-0003-4422-018X>)

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

Repository https://cran.r-universe.dev

Date/Publication 2026-10-30 19:31:37 UTC

RemoteUrl https://github.com/cran/LDAShiny

RemoteRef HEAD

RemoteSha 89723b276438297e9c51adc0222245d9ba3d6f71

Contents

mod_about_server 2
mod_about_ui 3
mod_import_bibliometrics_server 3
mod_import_bibliometrics_ui 4
mod_inference 4
mod_lda_train 5
mod_preprocess 5
mod_sensitivity 6
mod_trend 7
run_LDAShiny 7

Index **9**

`mod_about_server` *About Server*

Description

About Server

Usage

```
mod_about_server(id)
```

Arguments

`id`	Shiny module id

mod_about_ui *About UI*

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 containing: * ‘data‘ - cleaned combined dataset

mod_import_bibliometrics_ui *Import Bibliometrics UI*

Description

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

Usage

```
mod_import_bibliometrics_ui(id)
```

Arguments

id	Shiny module ID.

Value

A Shiny UI element.

mod_inference *LDA Inference Module*

Description

UI and server logic for LDA model selection. Runs multiple LDA models for a range of k and computes coherence to determine the optimal number of topics.

Usage

```
themes_gg

mod_inference_ui(id)

mod_inference_server(id, dtm_reactive)
```

Arguments

id	Shiny module id.
dtm_reactive	Reactive expression returning a dgCMatrix DTM.

Format

An object of class list of length 9.

Value

A list with: * model_list : reactive list of fitted LDA models for each k * coherence_mat : reactive data.frame with columns k and coherence * k_optimal : reactive integer, best k (max coherence)

---

mod_lda_train *Final LDA Model Training Module*

Description

Trains the final LDA model using the processed DTM. The number of topics (K) is automatically initialized with the optimal K found in the inference module, but the user can manually modify it.

Usage

```
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 expression returning the DTM (dgCMatrix)
k_optimal_reactive	Reactive expression returning the optimal K (integer)
bd_reactive	Reactive expression with the original metadata (Scopus/WoS)

Value

A list of reactive values containing the LDA model outputs.

---

mod_preprocess *Text Preprocessing Module for LDAShiny*

Description

Provides UI and server functions to clean, tokenize, stem, remove stopwords, build a document-term matrix (DTM), filter sparse terms, and export intermediate objects for topic modeling.

Usage

```
mod_preprocess_ui(id)

mod_preprocess_server(id, data_reactive)
```

Arguments

id	module id
data_reactive	reactive containing the input data frame

`mod_sensitivity` *Stability & Sensitivity Analysis Module*

Description

Evaluates how robust the chosen K is by: 1. Training LDA with multiple random seeds for candidate K values and measuring coherence variability (stability). 2. Computing Adjusted Rand Index (ARI) between seed-pairs to quantify assignment reproducibility. 3. Testing whether the winning K changes under alternative weight schemes (sensitivity of the composite score).

Usage

```
mod_sensitivity_ui(id)

mod_sensitivity_server(
  id,
  dtm_reactive,
  coherence_mat_reactive,
  k_optimal_reactive
)
```

Arguments

id	Shiny module id
dtm_reactive	Reactive expression returning the DTM (dgCMatrix)
coherence_mat_reactive	Reactive returning data.frame(k, coherence)
k_optimal_reactive	Reactive returning the best K (integer)

Value

A list of reactive values containing stability and sensitivity results.

mod_trend *Topic Trend Analysis Module*

Description

Performs linear regression analysis on topic intensity (Gamma mean) over years to classify topics as increasing (HOT), decreasing (COLD), or stable (EQUAL) trends.

Usage

```
mod_trend_ui(id)

mod_trend_server(id, df_year_topic_reactive, lda_model_reactive)
```

Arguments

`id`	Shiny module id
`df_year_topic_reactive`	Reactive expression returning the data frame with Topic, year, and Intensity (mean Gamma) from mod_lda_train.
`lda_model_reactive`	Reactive expression returning the final LDA model (passed for completeness, but not used for labels).

run_LDAShiny *Run the LDAShiny Application*

Description

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

Usage

```
run_LDAShiny(max_upload_mb = 2500, ...)
```

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

∗ **datasets**
mod_inference, 4

mod_about_server, 2
mod_about_ui, 3
mod_import_bibliometrics_server, 3
mod_import_bibliometrics_ui, 4
mod_inference, 4
mod_inference_server *(*mod_inference*)*, 4
mod_inference_ui *(*mod_inference*)*, 4
mod_lda_train, 5
mod_lda_train_server *(*mod_lda_train*)*, 5
mod_lda_train_ui *(*mod_lda_train*)*, 5
mod_preprocess, 5
mod_preprocess_server *(*mod_preprocess*)*, 5
mod_preprocess_ui *(*mod_preprocess*)*, 5
mod_sensitivity, 6
mod_sensitivity_server *(*mod_sensitivity*)*, 6
mod_sensitivity_ui *(*mod_sensitivity*)*, 6
mod_trend, 7
mod_trend_server *(*mod_trend*)*, 7
mod_trend_ui *(*mod_trend*)*, 7

run_LDAShiny, 7

themes_gg *(*mod_inference*)*, 4

with_golem_options, 7