Package 'revtools'

March 10, 2018

Version 0.2.2

Date 2018-03-09

Title Tools to Support Evidence Synthesis

Author Martin J. Westgate <martinjwestgate@gmail.com>

Maintainer Martin J. Westgate <martinjwestgate@gmail.com>

Description Researchers commonly need to summarize scientific information, a process known as 'evidence synthesis'. The first stage of a synthesis process (such as a systematic review or meta-analysis) is to download a list of references from academic search engines such as 'Web of Knowledge' or 'Scopus'. This information can be sorted manually (the traditional approach to systematic review), or the user can draw on tools from machine learning to help them visualise patterns in the corpus. 'revtools' uses topic models to render ordinations of text drawn from article titles, keywords and abstracts, and allows the user to interactively select or exclude individual references, words or topics. 'revtools' does not currently provide tools for analysis of data drawn from those references, features that are available in other packages such as 'metagear' or 'metafor'.

Depends R (>= 3.4.0)

Imports ade4, plotly, shiny, shinydashboard, SnowballC, stringdist, tm, topicmodels, viridisLite, methods, modeltools

License GPL-2

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-03-10 16:50:39 UTC

R topics documented:

`revtools-package` *revtools: Tools to support reviews and evidence synthesis*

Description

Researchers commonly need to summarize scientific information, a process known as 'evidence synthesis'. The first stage of a synthesis process (such as a systematic review or meta-analysis) is to download a list of references from academic search engines such as 'Web of Knowledge' or 'Scopus'. This information can be sorted manually (the traditional approach to systematic review), or the user can draw on tools from machine learning to help them visualise patterns in the corpus. `revtools` uses topic models to render ordinations of text drawn from article titles, keywords and abstracts, and allows the user to interactively select or exclude individual references, words or topics. `revtools` does not currently provide tools for analysis of data drawn from those references, features that are available in other packages such as `metagear` or `metafor`.

Functions

Import & export

- `read_bibliography` Import bibliographic data
- `write_bibliography` Export bibliographic data

Data storage and manipulation

- `bibliography-class` Format for storing bibliographic data
- `bibliography-methods` Print, summary, as.bibliography, as.data.frame and [methods for class 'bibliography'
- `review_info-class` Format for storing data from start_review_window
- `review_info-methods` summary methods for class review_info

Duplicate detection

- `find_duplicates` Locate potentially duplicated references
- `extract_unique_references` return a data.frame with only 'unique' references

Topic modelling and visualisation

- make_DTM Construct a Document-Term Matrix from bibliographic data
- run_LDA Wrapper function for topic models
- start_review_window Launch a Shiny app for reference sorting

---

avian_ecology_bibliography *Bibliographic data from 20 papers on avian ecology*

Description

This dataset lists basic information (title, authors, keywords etc.) for 20 scientific articles on avian ecology, stored in .ris format.

Usage

```
example_bibliography
```

Format

A list of length 20, containing lists of named attributes for each article.

Source

Originally downloaded from Scopus.

Examples

```
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)
summary(x)
```

---

bibliography-class *Description of class 'bibliography'*

Description

Class 'bibliography' is an S3 class designed to store data from common bibliographic formats in a standard way. It is a nested list format; each object is a list containing multiple references, where each reference is a list with information on author, journal etc. Because different formats code their information differently, class bibliography uses bib-like headings to ensure that reference names give a meaningful description of its' content (i.e. 'author' instead of 'AU'). This means that an .ris or medline and pubmed files have their tags converted on import by read_bibliography, while .bib tags are not altered.

slots

Class 'bibliography' will import all information given in a file, and will attempt to assign it to a sensible heading. Possible entry names are (in this order):

- **type** tag 'TY'
- **author** tags 'AU' or 'A' followed by 1:5
- **year** tags 'PY'or 'Y1'
- **title** tags 'TI' or 'T1'
- **journal** tags 'JO', 'T2', 'T3', 'SO', 'JT', 'JF' or 'JA'
- **volume** tag 'VL'
- **issue** tag 'IS'
- **pages** one or more of tags 'EP', 'BP' or 'SP'
- **abstract** tags 'AB' or 'N2'
- **keywords** tags 'KW' or 'DE'
- **doi** tag 'DO'
- **call** tag 'CN'
- **issn** tag 'SN'
- **url** tag 'UR'
- **accession** tag 'AN'
- **institution** tag 'CY'
- **publisher** tag 'PB'
- **pubplace** tag 'PP'
- **address** tag 'AD'
- **editor** tag 'ED'
- **edition** tag 'ET'
- **language** tag 'LA'
- **further_info** any unallocated information

`bibliography-methods` *Methods for class 'bibliography'*

Description

This is a small number of standard methods for interacting with class 'bibliography'. More may be added later.

Usage

```
as.bibliography(x, ...)
## S3 method for class 'bibliography'
as.data.frame(x, ...)
## S3 method for class 'bibliography'
x[n]
## S3 method for class 'bibliography'
print(x, n, ...)
## S3 method for class 'bibliography'
summary(object, ...)
```

Arguments

x	An object of class 'bibliography'
object	An object of class 'bibliography'
n	Number of items to select/print
...	Any further information

Examples

```
# import some data
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)

# basic descriptions
summary(x)
print(x)
x[1]

# conversion to and from data.frame
y<-as.data.frame(x)
x_new<-as.bibliography(y)
```

---

extract_unique_references

Create a de-duplicated data.frame

Description

Take a data.frame of bibliographic information showing potential duplicates (as returned by `find_duplicates`), and return a data.frame of unique references

Usage

```
extract_unique_references(x, show_source=FALSE)
```

Arguments

x	a data.frame as returned by find_duplicates
show_source	If x contains a column named 'source', selecting TRUE will cause the output to contain extra columns. Specifically, each added column will be named for a unique value of x$source, and each row will list the source(s) that contained that reference. Defaults to FALSE

Value

a data.frame containing basic information for each reference (row)

Note

This function creates a simplified version of that given by find_duplicates, by extracting the first reference from each group of unique references. There is no additional processing to ensure this is the 'best' reference from that list.

See Also

find_duplicates.

Examples

```
# import data
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-as.data.frame(read_bibliography(file_location))

# generate duplicated references (for example purposes)
x_duplicated<-rbind(x, x[1:5,])

# locate and extract unique references
x_check<-find_duplicates(x_duplicated)
x_unique<-extract_unique_references(x_check)
```

---

find_duplicates *Locate duplicated references within a data.frame*

Description

Identify potential duplicates within a data.frame containing title, journal and year data for each reference. Such a data.frame can be created by calling as.data.frame on an object of class bibliography (e.g. as returned by read_bibliography()).

Usage

```
find_duplicates(x)
```

Arguments

x	a data.frame containing title, journal and year data for each reference

Value

a data.frame with the same columns as the initial data, with a numeric variable named 'group'; rows with the same value are probable duplicates.

Note

This function has a few odd features. It uses stringdist to locate article titles that are similar to one another, rather than exact matching. This is done within a while loop; in each run of the loop, rows that have similar journal titles or publication years are checked for potential duplication, while rows with missing values of these variables are tested in every run. This makes the code fairly comprehensive at the cost of speed.

A method that tests string distances between all pairs would locate more duplicates, but would be substantially slower. Conversely, a method that split the dataset into mutually exclusive groups could be vectorized and would be correspondingly faster, but as the identity of these groups is inherently ambigious, it would probably reduce the hit rate.

Examples

```
# import data
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-as.data.frame(read_bibliography(file_location))

# generate then locate some 'fake' duplicates
x_duplicated<-rbind(x, x[1:5,])
x_check<-find_duplicates(x_duplicated)
# returns a data.frame with an added 'group' column
```

---

make_DTM *Construct a document-term matrix (DTM)*

Description

Takes bibliographic data and converts it to a DTM for passing to topic models.

Usage

```
make_DTM(x, stop_words)
```

Arguments

x	an object of class bibliography or data.frame containing bibliographic data
stop_words	optional vector of strings, listing terms to be removed from the DTM prior to analysis

Value

An object of class 'matrix', listing the terms (columns) present in each reference (rows)

Note

This is primarily intended to be called internally by `start_review_window()`, but is made available for users to generate their own topic models with the same properties as those in revtools. It bascially takes any words in the title, keywords and abstracts of the supplied references, and uses them to construct a DTM.

This function uses some standard tools like stemming, converting words to lower case, and removal of numbers or punctuation. It also replaces stemmed words with the most common full word, which doesn't affect the calculations, but makes the resulting analyses easier to interpret. It doesn't use part-of-speech tagging.

Examples

```
# import some data
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)

# construct a document-term matrix
# note: this can take a long time to run for large datasets
x_DTM<-make_DTM(x)
dim(x_DTM) # 20 articles, 1069 words
```

---

read_bibliography *Import bibliographic data*

Description

Import standard formats from academic search engines and referencing software.

Usage

```
read_bibliography(x, path)
```

Arguments

x	Filename of a bibliographic file. Supported formats include .ris, .bib, medline (.nbib) or web of science (.ciw)
path	Path to specified file

Value

Returns an object of class `bibliography`, which is a list where each entry is a list containing data on each reference. This function auto-detects document formatting, meaning that specified file formats are ignored except to locate the file.

Examples

```
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)
summary(x)
```

`review_info-class` *Description of class 'review_info'*

Description

Class 'review' is an S3 class designed to store data from the shiny app launched by `start_review_window`. This is important because it stores the many decisions that users might make about inclusion or exclusion of individual references from the corpus, information that will be needed for later stages of the review. Class review_info can also be passed to `start_review_window` to continue working on a previously altered dataset.

slots

Class 'review_info' has five slots containing the following information:

- **info** duplicate of data passed to `start_review_window`
- **dtm** document-term matrix, created by `make_DTM`
- **model** most recent topic model
- **plotinfo** data listing points to be plotted in `start_review_window`
- **infostore** data listing decisions about inclusion/exclusion of references, words or topics

---

`review_info-methods)` *Methods for class 'review_info'*

Description

Tools to display useful information on class `review_info`.

Usage

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

Arguments

`object`	An object of class 'review_info'
`...`	Any further information

Value

Prints useful information to the workspace.

Note

Class `review_info` is a format for exporting large quantities of data during reviews. It is typically stored within a .rds file in the working directory. When re-imported to R using `readRDS`, this file will contain an object of class `review_info`.

`run_LDA` *Calculate a topic model*

Description

Run a topic model using either `LDA` or `CTM` from the `topicmodels` package.

Usage

```
run_LDA(x, topic_model, n_topics, iterations)
```

Arguments

`x`	a Document Term Matrix (DTM)
`topic_model`	string specififying the type of model to run. Either 'lda' (the default) or 'ctm'.
`n_topics`	Number of topics to calculate
`iterations`	The number of iterations. Only relevant for LDA.

Value

A topic model with the specified parameters.

Note

This is a basic wrapper function designed to allow consistent specification of model parameters within `shiny` apps. It doesn't do anything very clever.

Examples

```
# import data
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)

# run a topic model based on these data
# note: the following lines can take a very long time to run, especially for large datasets!
x_DTM<-make_DTM(x)
## Not run: x_LDA<-run_LDA(x_DTM, "lda", 5, 5000)
```

start_review_window *Interactive visualisation of bibliographic data*

Description

Draw an interactive plot of a bibliography. Ordinations are calculated using LDA (library "topicmodels") and are displayed using shiny and plotly.

Usage

```
start_review_window(x, remove_words)
```

Arguments

x	Bibliographic data, in one of three formats: a list returned by read_bibliography(); a data.frame; or a previously saved output from start_review_window (class("review_info")).
remove_words	vector of words to be removed from consideration by the Topic Model. If none are given, start_review_window will use tm::stopwords(). Note that this vector will be converted to lower case before processing, so the algorithm is not case sensitive.

Value

This function launches a Shiny app in the users' default browser.

The display space is divided into three parts. From left to right, these are the sidebar; the plot window; and the selection panel.

The sidebar shows a series of drop-down menus that can be used to customize or recalculate the central plot. It can be minimized when not in use. Note that the default settings for LDA (5 topics, 1000 iterations) prioritize speed over reliability - higher numbers of iterations will give more reliable results.

The plot window shows an ordination of article weights calculated using LDA, with articles colored by their highest-weighted topic. Hovering over a point shows the title and abstract below the plot; clicking allows selection or deselection of that article (and optionally displays co-authorship data). Selecting a region of the plot and clicking zooms on the selected region; double-clicking without selecting a region returns the plot to its full extent.

The selection panel gives information on progress in selecting/deselecting articles. It also contains windwos for displaying topic-level infromation and article abstracts. All boxes in this panel can be minimized when not required.

Upon completion, the user can export information to a .csv or .rda file (saved to the working directory) using the 'Save' tab.

Examples

```
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)
## Not run: start_review_window(x)
```

`write_bibliography` *Export imported bibliographic data as .bib or .ris formats*

Description

Basic function to export bibliographic information for use in other programs. Work in progress. Very little error checking or advanced formatting in this version

Usage

```
write_bibliography(x, filename, format="ris")
```

Arguments

`x`	An object of class 'bibliography', such as imported using read_bibliography
`filename`	Name of the exported file. Should ideally match 'format', but this is not enforced
`format`	Format of the exported file. Should be either "ris" (default) or "bib"

Value

exports results as a .ris or .bib file.

Examples

```
file_location<-system.file("extdata", "avian_ecology_bibliography.ris", package="revtools")
x<-read_bibliography(file_location)

# export a subset of entries as a new file
write_bibliography(x[1:5],
  filename=paste0(tempdir(), "/x_out.ris"),
  format="ris")
```

Index