Package 'RNewsflow'

February 21, 2016

Type Package

Title Tools for Analyzing Content Homogeneity and News Diffusion using Computational Text Analysis

Version 1.0

Date 2016-02-14

Author Kasper Welbers & Wouter van Atteveldt

Maintainer Kasper Welbers <kasperwelbers@gmail.com>

Description A collection of tools for measuring the similarity of news content and tracing the flow of (news) messages over time and across media.

License MIT + file LICENSE

Depends tm, igraph, Matrix

Imports plyr, slam, scales, wordcloud, data.table, methods

RoxygenNote 5.0.1

Suggests knitr, rmarkdown, RTextTools

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-21 19:51:39

R topics documented:

delete.duplicates 2
directed.network.plot 3
docnet 5
document.network 5
document.network.plot 6
documents.compare 7
dtm 8
filter.window 9
meta 10

network.aggregate 10
newsflow.compare 12
only.first.match 13
show.window 14
term.day.dist 15

Index **17**

`delete.duplicates` *Delete duplicate (or similar) documents from a document term matrix*

Description

Delete duplicate (or similar) documents from a document term matrix. Duplicates are defined by: having high content similarity, occuring within a given time distance and being published by the same source.

Usage

```
delete.duplicates(dtm, meta, id.var = "document_id", date.var = "date",
  source.var = "source", hour.window = c(-24, 24), measure = "cosine",
  similarity = 1, keep = "first", tf.idf = FALSE)
```

Arguments

`dtm`	A document-term matrix in the tm DocumentTermMatrix class. It is recommended to weight the DTM beforehand, for instance using weightTfIdf.
`meta`	A data.frame where rows are documents and columns are document meta information. Should contain 3 columns: the document name/id, date and source. The name/id column should match the document names/ids of the edgelist, and its label is specified in the ‘id.var‘ argument. The date column should be intepretable with as.POSIXct, and its label is specified in the ‘date.var‘ argument. The source column is specified in the ‘date.var‘ argument.
`id.var`	The label for the document name/id column in the ‘meta‘ data.frame. Default is "document_id"
`date.var`	The label for the document date column in the ‘meta‘ data.frame . default is "date"
`source.var`	The label for the document date column in the ‘meta‘ data.frame . default is "source"
`hour.window`	A vector of length 2, in which the first and second value determine the left and right side of the window, respectively. By default c(-24,24), which compares each document to all other documents within a 24 hour time distance.
`measure`	the measure that should be used to calculate similarity/distance/adjacency. Currently supports the symmetrical measure "cosine" (cosine similarity), and the assymetrical measures "overlap_pct" (percentage of term scores in the document that also occur in the other document).

similarity	a threshold for similarity. Documents of which similarity is equal or higher are deleted
keep	A character indicating whether to keep the 'first' or 'last' published of duplicate documents.
tf.idf	if TRUE, weight the dtm with tf.idf before comparing documents. The original (non-weighted) DTM is returned.

Details

Note that this can also be used to delete "updates" of articles (e.g., on news sites, news agencies). This should be considered if the temporal order of publications is relevant for the analysis.

Value

A dtm with the duplicate documents deleted

Examples

```
data(dtm)
data(meta)

## example with very low similarity threshold (normally not recommended!)
dtm2 = delete.duplicates(dtm, meta, similarity = 0.5, keep='first', tf.idf = TRUE)
```

---

directed.network.plot *A wrapper for plot.igraph for visualizing directed networks.*

Description

This is a convenience function for visualizing directed networks with edge labels using plot.igraph. It was designed specifically for visualizing aggregated document similarity networks in the RNewsflow package, but works with any network in the igraph class.

Usage

```
directed.network.plot(g, weight.var = "from.Vprop", weight.thres = NULL,
  delete.isolates = FALSE, vertex.size = 30, vertex.color = "lightblue",
  vertex.label.color = "black", vertex.label.cex = 0.7,
  edge.color = "grey", show.edge.labels = TRUE,
  edge.label.color = "black", edge.label.cex = 0.6, edge.arrow.size = 1,
  layout = igraph::layout.davidson.harel, ...)
```

Arguments

g	A network/graph in the igraph class
weight.var	The edge attribute that is used to specify the edges
weight.thres	A threshold for weight. Edges below the threshold are ignored
delete.isolates	If TRUE, isolates (i.e. vertices without edges) are ignored.
vertex.size	The size of the verticex/nodes. Defaults to 30. Can be a vector with values per vertex.
vertex.color	Color of vertices/nodes. Default is lightblue. Can be a vector with values per vertex.
vertex.label.color	Color of labels for vertices/nodes. Defaults to black. Can be a vector with values per vertex.
vertex.label.cex	Size of the labels for vertices/nodes. Defaults to 0.7. Can be a vector with values per vertex.
edge.color	Color of the edges. Defaults to grey. Can be a vector with values per edge.
show.edge.labels	Logical. Should edge labels be displayed? Default is TRUE.
edge.label.color	Color of the edge labels. Defaults to black. Can be a vector with values per edge.
edge.label.cex	Size of the edge labels. Defaults to 0.6. Can be a vector with values per edge.
edge.arrow.size	Size of the edge arrows. Defaults to 1. Can only be set globally (igraph might update this at some point)
layout	The igraph layout used to plot the network. Defaults to layout.davidson.harel
...	Arguments to be passed to the plot.igraph function.

Value

Nothing

Examples

```
data(docnet)
aggdocnet = network.aggregate(docnet, by='source')
directed.network.plot(aggdocnet, weight.var = 'to.Vprop', weight.thres = 0.2)
```

docnet — *Document similarity network for one news agency, and the print and online editions of two newspapers*

Description

Document similarity network for one news agency, and the print and online editions of two newspapers

Usage

```
data(docnet)
```

Format

docnet: A network/graph in the igraph class as created with document.network or newsflow.compare.

document.network — *Create a document similarity network*

Description

Combines document similarity data (d) with document meta data (meta) into an igraph network/graph.

Usage

```
document.network(d, meta, id.var = "document_id", date.var = "date",
  min.similarity = 0)
```

Arguments

d	A data.frame with three columns, that represents an edgelist with weight values. The first and second column represent the names/ids of the 'from' and 'to' documents/vertices. The third column represents the similarity score. Column names are ignored
meta	A data.frame where rows are documents and columns are document meta information. Should at least contain 2 columns: the document name/id and date. The name/id column should match the document names/ids of the edgelist, and its label is specified in the 'id.var' argument. The date column should be intepretable with as.POSIXct, and its label is specified in the 'date.var' argument.
id.var	The label for the document name/id column in the 'meta' data.frame. Default is "document_id"
date.var	The label for the document date column in the 'meta' data.frame . default is "date"
min.similarity	For convenience, ignore all edges where the weight is below 'min.similarity'. Default is zero.

Details

This function is mainly offered to mimic the output of the newsflow.compare function when using imported document similarity data. This way the functions for transforming, aggregating and visualizing the document similarity data can be used.

Value

A network/graph in the igraph class

Examples

```
d = data.frame(x = c(1,1,1,2,2,3),
               y = c(2,3,5,4,5,6),
               v = c(0.3,0.4,0.7,0.5,0.2,0.9))

meta = data.frame(document_id = 1:8,
                  date = seq.POSIXt(from = as.POSIXct('2010-01-01 12:00:00'),
                         by='hour', length.out = 8),
                  medium = c(rep('Newspapers', 4), rep('Blog', 4)))

g = document.network(d, meta)

igraph::get.data.frame(g, 'both')
igraph::plot.igraph(g)
```

---

document.network.plot *Visualize (a subcomponent) of the document similarity network*

Description

Visualize (a subcomponent) of the document similarity network

Usage

```
document.network.plot(g, date.attribute = "date",
  source.attribute = "source", subcomp_i = NULL, dtm = NULL,
  sources = NULL, only.outer.date = FALSE,
  date.format = "%Y-%m-%d %H:%M", margins = c(5, 8, 1, 13),
  isolate.color = NULL, source.loops = TRUE, ...)
```

Arguments

g	A document similarity network, as created with newsflow.compare or document.network
date.attribute	The label of the vertex/document date attribute. Default is "date"
source.attribute	The label of the vertex/document source attribute. Default is "source"

subcomp_i	Optional. If an integer is given, the network is decomposed into subcomponents (i.e. unconnected components) and only the i-th component is visualized.
dtm	Optional. If a document-term matrix that contains the documents in g is given, a wordcloud with the most common words of the network is added.
sources	Optional. Use a character vector to select only certain sources
only.outer.date	If TRUE, only the labels for the first and last date are reported on the x-axis
date.format	The date format of the date labels (see format.POSIXct)
margins	The margins of the network plot. The four values represent bottom, left, top and right margin.
isolate.color	Optional. Set a custom color for isolates
source.loops	If set to FALSE, all edges between vertices/documents of the same source are ignored.
...	Additional arguments for the network plotting function plot.igraph

Value

Nothing.

Examples

```
data(docnet)
data(dtm)

docnet_comps = igraph::decompose.graph(docnet) # get subcomponents

# subcomponent 1
document.network.plot(docnet_comps[[1]])

# subcomponent 2 with wordcloud
document.network.plot(docnet_comps[[2]], dtm=dtm)

# subcomponent 3 with additional arguments passed to plot.igraph
document.network.plot(docnet_comps[[3]], dtm=dtm, vertex.color='red')
```

---

documents.compare *Compare the documents in two corpora/dtms*

Description

Compare the documents in corpus dtm.x with reference corpus dtm.y.

Usage

```
documents.compare(dtm, dtm.y = NULL, measure = "cosine",
  min.similarity = 0, n.topsim = NULL, return.zeros = FALSE)
```

Arguments

dtm	A document-term matrix in the tm DocumentTermMatrix class. It is recommended to weight the DTM beforehand, for instance using weightTfIdf.
dtm.y	Optional. If given, documents from dtm will only be compared to the documents in dtm.y
measure	the measure that should be used to calculate similarity/distance/adjacency. Currently supports the symmetrical measure "cosine", for cosine similarity. Also supports assymetrical measures "percentage.from" and "percentage.to" for the percentage of overlapping terms (term scores taken into account). Here "percentage.from" gives the percentage of the document that is compared to the other, whereas "percentage.to" gives the percentage of the document to which is compared.
min.similarity	a threshold for similarity. lower values are deleted. Set to 0 by default.
n.topsim	An alternative or additional sort of threshold for similarity. Only keep the [n.topsim] highest similarity scores for x. Can return more than [n.topsim] similarity scores in the case of duplicate similarities.
return.zeros	If true, all comparison results are returned, including those with zero similarity (rarely usefull and problematic with large data)

Details

The calculation of document similarity is performed using a vector space model approach. Innerproduct based similarity measures are used, such as cosine similarity. It is recommended to weight the DTM beforehand, for instance using Term frequency-inverse document frequency (tf.idf)

Value

A data frame with pairs of documents and their similarities.

Examples

```
data(dtm)

comp = documents.compare(dtm, min.similarity=0.4)
head(comp)
```

---

dtm *Document Term Matrix for one news agency, and the print and online editions of two newspapers*

Description

Document Term Matrix for one news agency, and the print and online editions of two newspapers

Usage

```
data(dtm)
```

Format

dtm: A document term matrix in the 'DocumentTermMatrix' class of the tm package.

`filter.window` *Filter edges from the document similarity network based on hour difference*

Description

The 'filter.window' function can be used to filter the document pairs (i.e. edges) using the 'hour.window' parameter, which works identical to the 'hour.window' parameter in the 'newsflow.compare' function. In addition, the 'from.vertices' and 'to.vertices' parameters can be used to select the vertices (i.e. documents) for which this filter is applied.

Usage

```
filter.window(g, hour.window, to.vertices = NULL, from.vertices = NULL)
```

Arguments

`g`	A document similarity network, as created with newsflow.compare or document.network
`hour.window`	A vector of length 2, in which the first and second value determine the left and right side of the window, respectively. For example, c(-10, 36) will compare each document to all documents between the previous 10 and the next 36 hours.
`to.vertices`	A filter to select the vertices 'to' which an edge is filtered. For example, if 'V(g)$sourcetype == "newspaper"' is used, then the hour.window filter is only applied for edges 'to' newspaper documents (specifically, where the sourcetype attribute is "newspaper").
`from.vertices`	A filter to select the vertices 'from' which an edge is filtered. Works identical to 'to.vertices'.

Details

It is recommended to use the show.window function to verify whether the hour windows are correct according to the assumptions and focus of the study.

Value

A network/graph in the igraph class

Examples

```
data(docnet)
show.window(docnet, to.attribute = 'source') # before filtering

docnet = filter.window(docnet, hour.window = c(0.1,24))

docnet = filter.window(docnet, hour.window = c(6,36),
                       to.vertices = V(docnet)$sourcetype == 'Print NP')

show.window(docnet, to.attribute = 'sourcetype') # after filtering per sourcetype
show.window(docnet, to.attribute = 'source') # after filtering per source
```

`meta` *Meta information for one news agency, and the print and online editions of two newspapers*

Description

Meta information for one news agency, and the print and online editions of two newspapers

Usage

```
data(meta)
```

Format

meta: A data.frame containing the meta information for each document.

`network.aggregate` *Aggregate the edges of a network by vertex attributes*

Description

This function offers a versatile way to aggregate the edges of a network based on the vertex attributes. Although it was designed specifically for document similarity networks, it can be used for any network in the igraph class.

Usage

```
network.aggregate(g, by = NULL, by.from = by, by.to = by,
  edge.attribute = "weight", agg.FUN = mean, return.df = FALSE)
```

Arguments

g	A network/graph in the igraph class
by	A character string indicating the vertex attributes by which the edges will be aggregated.
by.from	Optionally, specify different vertex attributes to aggregate the 'from' side of edges
by.to	Optionally, specify different vertex attributes to aggregate the 'to' side of edges
edge.attribute	Select an edge attribute to aggregate using the function specified in 'agg.FUN'. Defaults to 'weight'
agg.FUN	The function used to aggregate the edge attribute
return.df	Optional. If TRUE, the results are returned as a data.frame. This can in particular be convenient if by.from and by.to are used.

Details

The first argument is the network (in the 'igraph' class). The second argument, for the 'by' parameter, is a character vector to indicate one or more vertex attributes based on which the edges are aggregated. Optionally, the 'by' parameter can also be specified separately for 'by.from' and 'by.to'.

By default, the function returns the aggregated network as an igraph class. The edges in the aggregated network have five standard attributes. The 'edges' attribute counts the number of edges from the 'from' group to the 'to' group. The 'from.V' attribute shows the number of vertices in the 'from' group that matched with a vertex in the 'to' group. The 'from.Vprop attribute shows this as the proportion of all vertices in the 'from' group. The 'to.V' and 'to.Vprop' attributes show the same for the 'to' group.

In addition, one of the edge attributes of the original network can be aggregated with a given function. These are specified in the 'edge.attribute' and 'agg.FUN' parameters.

Value

A network/graph in the igraph class, or a data.frame if return.df is TRUE.

Examples

```
data(docnet)
aggdocnet = network.aggregate(docnet, by='sourcetype')
igraph::get.data.frame(aggdocnet, 'both')

aggdocdf = network.aggregate(docnet, by.from='sourcetype', by.to='source', return.df = TRUE)
head(aggdocdf)
```

`newsflow.compare` *Compare the documents in a dtm with a sliding window over time*

Description

Given a document-term matrix (DTM) and corresponding document meta data, calculates the document similarities over time using with a sliding window.

The meta data.frame should have a column containing document id's that match the rownames of the DTM (i.e. document names) and should have a column indicating the publication time. By default these columns should be labeled "document_id" and "date", but the column labels can also be set using the 'id.var' and 'date.var' parameters. Any other columns will automatically be included as document meta information in the output.

Usage

```
newsflow.compare(dtm, meta, id.var = "document_id", date.var = "date",
  hour.window = c(-24, 24), measure = "cosine", min.similarity = 0,
  n.topsim = NULL, only.from = NULL, only.to = NULL,
  return.zeros = FALSE, only.complete.window = TRUE)
```

Arguments

`dtm`	A document-term matrix in the tm DocumentTermMatrix class. It is recommended to weight the DTM beforehand, for instance using weightTfIdf.
`meta`	A data.frame where rows are documents and columns are document meta information. Should at least contain 2 columns: the document name/id and date. The name/id column should match the document names/ids of the edgelist, and its label is specified in the 'id.var' argument. The date column should be intepretable with as.POSIXct, and its label is specified in the 'date.var' argument.
`id.var`	The label for the document name/id column in the 'meta' data.frame. Default is "document_id"
`date.var`	The label for the document date column in the 'meta' data.frame . default is "date"
`hour.window`	A vector of length 2, in which the first and second value determine the left and right side of the window, respectively. For example, c(-10, 36) will compare each document to all documents between the previous 10 and the next 36 hours.
`measure`	the measure that should be used to calculate similarity/distance/adjacency. Currently supports the symmetrical measure "cosine" (cosine similarity), and the assymetrical measures "overlap_pct" (percentage of term scores in the document that also occur in the other document).
`min.similarity`	a threshold for similarity. lower values are deleted. Set to 0.1 by default.
`n.topsim`	An alternative or additional sort of threshold for similarity. Only keep the [n.topsim] highest similarity scores for x. Can return more than [n.topsim] similarity scores in the case of duplicate similarities.

`only.from` A vector with names/ids of documents (dtm rownames), or a logical vector that matches the rows of the dtm. Use to compare only these documents to other documents.

`only.to` A vector with names/ids of documents (dtm rownames), or a logical vector that matches the rows of the dtm. Use to compare other documents to only these documents.

`return.zeros` If true, all comparison results are returned, including those with zero similarity (rarely usefull and problematic with large data)

`only.complete.window` if True, only compare articles (x) of which a full window of reference articles (y) is available. Thus, for the first and last [window.size] days, there will be no results for x.

Details

The calculation of document similarity is performed using a vector space model approach. Inner-product based similarity measures are used, such as cosine similarity. It is recommended to weight the DTM beforehand, for instance using Term frequency-inverse document frequency (tf.idf)

Value

A network/graph in the igraph class

Examples

```
data(dtm)
data(meta)

dtm = tm::weightTfIdf(dtm)
g = newsflow.compare(dtm, meta, hour.window = c(0.1, 36))

vcount(g) # number of documents, or vertices
ecount(g) # number of document pairs, or edges

head(igraph::get.data.frame(g, 'vertices'))
head(igraph::get.data.frame(g, 'edges'))
```

---

only.first.match *Transform document network so that each document only matches the earliest dated matching document*

Description

Transforms the network so that a document only has an edge to the earliest dated document it matches within the specified time window[^duplicate].

Usage

```
only.first.match(g)
```

Arguments

g	A document similarity network, as created with newsflow.compare or document.network

Details

If there are multiple earliest dated documents (that is, having the same publication date) then edges to all earliest dated documents are kept.

Value

A network/graph in the igraph class

Examples

```
data(docnet)

subcomp1 = igraph::decompose.graph(docnet)[[2]]
subcomp2 = only.first.match(subcomp1)

igraph::get.data.frame(subcomp1)
igraph::get.data.frame(subcomp2)

graphics::par(mfrow=c(2,1))
document.network.plot(subcomp1, main='All matches')
document.network.plot(subcomp2, main='Only first match')
graphics::par(mfrow=c(1,1))
```

---

show.window *Show time window of document pairs*

Description

This function aggregates the edges for all combinations of attributes specified in ‘from.attribute‘ and ‘to.attribute‘, and shows the minimum and maximum hour difference for each combination.

Usage

```
show.window(g, to.attribute = NULL, from.attribute = NULL)
```

Arguments

g	A document similarity network, as created with newsflow.compare or document.network
to.attribute	The vertex attribute to aggregate the ‘to‘ group of the edges
from.attribute	The vertex attribute to aggregate the ‘from‘ group of the edges

Details

The filter.window function can be used to filter edges that fall outside of the intended time window.

Value

A data.frame showing the left and right edges of the window for each unique group.

Examples

```
data(docnet)
show.window(docnet, to.attribute = 'source')
show.window(docnet, to.attribute = 'sourcetype')
show.window(docnet, to.attribute = 'sourcetype', from.attribute = 'sourcetype')
```

---

term.day.dist *Calculate statistics for term occurence across days*

Description

Calculate statistics for term occurence across days

Usage

```
term.day.dist(dtm, meta, id.var = "document_id", date.var = "date")
```

Arguments

dtm	A document-term matrix in the tm DocumentTermMatrix class or a TsparseMatrix from the Matrix class (spMatrix)
meta	A data.frame where rows are documents and columns are document meta information. Should contain 2 columns: the document name/id and date. The name/id column should match the rownames (i.e. document names) of the DTM, and its label is specified in the ‘id.var‘ argument. The date column should be intepretable with as.POSIXct, and its label is specified in the ‘date.var‘ argument.
id.var	The label for the document name/id column in the ‘meta‘ data.frame. Default is "document_id"
date.var	The label for the document date column in the ‘meta‘ data.frame . default is "date"

Value

A data.frame with statistics for each term.

- freq: The number of times a term occurred
- doc.freq: The number of documents in which a term occured
- days.n: The number of days on which a term occured
- days.pct: The percentage of days on which a term occured
- days.entropy: The entropy of the distribution of term frequency across days
- days.entropy.norm: The normalized days.entropy, where 1 is a discrete uniform distribution

Examples

```
data(dtm)
data(meta)

tdd = term.day.dist(dtm, meta)
head(tdd)
tail(tdd)
```

Index

∗Topic **datasets**
docnet, 5
dtm, 8
meta, 10

as.POSIXct, *2*, *5*, *12*, *15*

delete.duplicates, 2
directed.network.plot, 3
docnet, 5
document.network, *5*, 5, *6*, *9*, *14, 15*
document.network.plot, 6
documents.compare, 7
DocumentTermMatrix, *2*, *8*, *12*, *15*
dtm, 8

filter.window, 9, *15*
format.POSIXct, *7*

igraph, *3–6*, *9–11*, *13, 14*

layout.davidson.harel, *4*

meta, 10

network.aggregate, 10
newsflow.compare, *5, 6*, *9*, 12, *14, 15*

only.first.match, 13

plot.igraph, *3, 4*, *7*

show.window, *9*, 14
spMatrix, *15*

term.day.dist, 15

weightTfIdf, *2*, *8*, *12*