Package ‘RTextTools’

October 12, 2022

Type Package
Title Automatic Text Classification via Supervised Learning
Version 1.4.3
Date 2020-04-24
Author Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun, Emiliano Grossman, Wouter van Atteveldt
Maintainer Loren Collingwood <loren.collingwood@gmail.com>
Depends R (>= 3.6.0), SparseM
Imports methods, randomForest, tree, nnet, tm, e1071, ipred, caTools, glmnet, tau
Description A machine learning package for automatic text classification that makes it simple for novice users to get started with machine learning, while allowing experienced users to easily experiment with different settings and algorithm combinations. The package includes eight algorithms for ensemble classification (svm, slda, boosting, bagging, random forests, glmnet, decision trees, neural networks), comprehensive analytics, and thorough documentation.
License GPL-3
URL http://www.rtexttools.com/
NeedsCompilation yes
Repository CRAN
Date/Publication 2020-04-26 01:10:02 UTC

R topics documented:

analytics-class ................................................................. 2
analytics_virgin-class ......................................................... 3
as.compressed.matrix .......................................................... 3
classify_model ................................................................. 4
classify_models ................................................................. 5
create_analytics ............................................................... 6
analytics-class

An S4 class containing the analytics for a classified set of documents. This includes a label summary, document summary, ensemble summary, and algorithm summary. This class is returned if `virgin=FALSE` in `create_container`.

Objects from the Class

Objects could in principle be created by calls of the form `new("analytics", ...)`. The preferred form is to have them created via a call to `create_analytics`.

Slots

- `label_summary` Object of class "data.frame": stores the analytics for each label, including the percent coded accurately and how much overcoding occurred
- `document_summary` Object of class "data.frame": stores the analytics for each document, including all available raw data associated with the learning process
- `algorithm_summary` Object of class "data.frame": stores precision, recall, and F-score statistics for each algorithm, broken down by label
- `ensemble_summary` Object of class "matrix": stores the accuracy and coverage for an n-algorithm ensemble scoring
analytics_virgin-class

An S4 class containing the analytics for a classified set of documents.

Description

An S4 class containing the analytics for a classified set of documents. This includes a label summary and a document summary. This class is returned if virgin=TRUE in create_container.

Objects from the Class

Objects could in principle be created by calls of the form new("analytics_virgin", ...). The preferred form is to have them created via a call to create_analytics.

Slots

- label_summary Object of class "data.frame": stores the analytics for each label, including how many documents were classified with each label
- document_summary Object of class "data.frame": stores the analytics for each document, including all available raw data associated with the learning process

Author(s)

Timothy P. Jurka <tpjurka@ucdavis.edu>

as.compressed.matrix

converts a tm DocumentTermMatrix or TermDocumentMatrix into a matrix.csr representation.

Description

Converts a DocumentTermMatrix or TermDocumentMatrix (package tm), Matrix (package Matrix), matrix.csr (SparseM), data.frame, or matrix into a matrix.csr representation to be used in the RTextTools functions.

Usage

as.compressed.matrix(DocumentTermMatrix)

Arguments

DocumentTermMatrix

A class of type DocumentTermMatrix or TermDocumentMatrix (package tm), Matrix (package Matrix), matrix.csr (SparseM), data.frame, or matrix.
classify_model

Value

A \texttt{matrix.csr} representation of the DocumentTermMatrix or TermDocumentMatrix (package tm), Matrix (package Matrix), matrix.csr (SparseM), data.frame, or matrix.

Author(s)

Timothy P. Jurka <tpjurka@ucdavis.edu>

classify_model \textit{makes predictions from a train_model()} object.

Description

Uses a trained model from the \texttt{train_model} function to classify new data.

Usage

classify_model(container, model, s=0.01, ...)

Arguments

container 
Class of type \texttt{matrix_container-class} generated by the \texttt{create_container} function.

model 
Slot for trained SVM, SLDA, boosting, bagging, RandomForests, glmnet, decision tree, neural network, or maximum entropy model generated by \texttt{train_model}.

s 
Penalty parameter lambda for \texttt{glmnet} classification.

... 
Additional parameters to be passed into the \texttt{predict} function of any algorithm.

Details

Only one model may be passed in at a time for classification. See \texttt{train_models} and \texttt{classify_models} to train and classify using multiple algorithms.

Value

Returns a \texttt{data.frame} of predicted codes and probabilities for the specified algorithm.

Author(s)

Loren Collingwood <loren.collingwood@gmail.com>, Timothy P. Jurka <tpjurka@ucdavis.edu>
classify_models

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english",
removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100,
virgin=FALSE)
svm_model <- train_model(container, "SVM")
svm_results <- classify_model(container, svm_model)

classify_models makes predictions from a train_models() object.

Description

Uses a trained model from the train_models function to classify new data.

Usage

classify_models(container, models, ...)

Arguments

container Class of type matrix_container-class generated by the create_container function.
models List of models to be used for classification generated by train_models.
... Other parameters to be passed on to classify_model.

Details

Use the list returned by train_models to use multiple models for classification.

Author(s)

Wouter Van Atteveldt <wouter@vanatteveldt.com>, Timothy P. Jurka <tpjurka@ucdavis.edu>

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english",
removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100,
virgin=FALSE)
models <- train_models(container, algorithms=c("RF","SVM"))
results <- classify_models(container, models)
create_analytics creates an object of class analytics given classification results.

Description

Takes the results from functions classify_model or classify_models and computes various statistics to help interpret the data.

Usage

create_analytics(container, classification_results, b=1)

Arguments

container Class of type matrix_container-class generated by the create_container function.

classification_results A cbind() of result objects returned by classify_model, or the object returned by classify_models.

b b-value for generating precision, recall, and F-scores statistics.

Value

Object of class analytics_virgin-class or analytics-class has either two or four slots respectively, depending on whether the virgin flag is set to TRUE or FALSE in create_container. They can be accessed using the @ operator for S4 classes (e.g. analytics@document_summary).

Author(s)

Timothy P. Jurka <tpjurka@ucdavis.edu>, Loren Collingwood <lorenc2@uw.edu>

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100, virgin=FALSE)
models <- train_models(container, algorithms=c("RF", "SVM"))
results <- classify_models(container, models)
analytics <- create_analytics(container, results)
create_container creates a container for training, classifying, and analyzing documents.

Description

Given a DocumentTermMatrix from the tm package and corresponding document labels, creates a container of class matrix_container-class that can be used for training and classification (i.e. train_model, train_models, classify_model, classify_models)

Usage

create_container(matrix, labels, trainSize=NULL, testSize=NULL, virgin)

Arguments

matrix A document-term matrix of class DocumentTermMatrix or TermDocumentMatrix from the tm package, or generated by create_matrix.
labels A factor or vector of labels corresponding to each document in the matrix.
trainSize A range (e.g. 1:1000) specifying the number of documents to use for training the models. Can be left blank for classifying corpora using saved models that don't need to be trained.
testSize A range (e.g. 1:1000) specifying the number of documents to use for classification. Can be left blank for training on all data in the matrix.
virgin A logical (TRUE or FALSE) specifying whether to treat the classification data as virgin data or not.

Value

A container of class matrix_container-class that can be passed into other functions such as train_model, train_models, classify_model, classify_models, and create_analytics.

Author(s)

Timothy P. Jurka <tpjurka@ucdavis.edu>, Loren Collingwood <loren.collingwood@gmail.com>

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100, virgin=FALSE)
create_ensembleSummary

creates a summary with ensemble coverage and precision.

Description

Creates a summary with ensemble coverage and precision values for an ensemble greater than the threshold specified.

Usage

create_ensembleSummary(document_summary)

Arguments

document_summary

The document_summary slot from the analytics-class generated by create_analytics.

Details

This summary is created in the create_analytics function. Note that a threshold value of 3 will return ensemble coverage and precision statistics for topic codes that had 3 or more (i.e. >=3) algorithms agree on the same topic code.

Author(s)

Loren Collingwood, Timothy P. Jurka

create_matrix

creates a document-term matrix to be passed into create_container().

Description

Creates an object of class DocumentTermMatrix from tm that can be used in the create_container function.

Usage

create_matrix(textColumns, language="english", minDocFreq=1, maxDocFreq=Inf, minWordLength=3, maxWordLength=Inf, ngramLength=1, originalMatrix=NULL, removeNumbers=FALSE, removePunctuation=TRUE, removeSparseTerms=0, removeStopwords=TRUE, stemWords=FALSE, stripWhitespace=TRUE, toLower=TRUE, weighting=weightTf)
create_matrix

Arguments

textColumns Either character vector (e.g. data$Title) or a cbind() of columns to use for training the algorithms (e.g. cbind(data$Title, data$Subject)).

language The language to be used for stemming the text data.

minDocFreq The minimum number of times a word should appear in a document for it to be included in the matrix. See package tm for more details.

maxDocFreq The maximum number of times a word should appear in a document for it to be included in the matrix. See package tm for more details.

minWordLength The minimum number of letters a word or n-gram should contain to be included in the matrix. See package tm for more details.

maxWordLength The maximum number of letters a word or n-gram should contain to be included in the matrix. See package tm for more details.

ngramLength The number of words to include per n-gram for the document-term matrix.

originalMatrix The original DocumentTermMatrix used to train the models. If supplied, will adjust the new matrix to work with saved models.

removeNumbers A logical parameter to specify whether to remove numbers.

removePunctuation A logical parameter to specify whether to remove punctuation.

removeSparseTerms A logical parameter to specify whether to remove sparse terms.

removeStopwords A logical parameter to specify whether to remove stopwords using the language specified in language.

stemWords A logical parameter to specify whether to stem words using the language specified in language.

stripWhitespace A logical parameter to specify whether to strip whitespace.

toLower A logical parameter to specify whether to make all text lowercase.

weighting Either weightTf or tm::weightTfIdf. See package tm for more details.

Author(s)

Timothy P. Jurka <tpjurka@ucdavis.edu>, Loren Collingwood <lorenc2@uw.edu>

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
create_precisionRecallSummary

creates a summary with precision, recall, and F1 scores.

Description

Creates a summary with precision, recall, and F1 scores for each algorithm broken down by unique label.

Usage

create_precisionRecallSummary(container, classification_results, b_value = 1)

Arguments

container Class of type matrix_container-class generated by the create_container function.

classification_results A cbind() of result objects returned by classify_model, or the object returned by classify_models.

b_value b-value for generating precision, recall, and F-scores statistics.

Author(s)

Loren Collingwood, Timothy P. Jurka

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"])), language="english",
removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100,
virgin=FALSE)
models <- train_models(container, algorithms=c("RF","SVM"))
results <- classify_models(container, models)
precision_recall_f1 <- create_precisionRecallSummary(container, results)
create_scoreSummary

create_scoreSummary creates a summary with the best label for each document.

Description

Creates a summary with the best label for each document, determined by highest algorithm certainty, and highest consensus (i.e. most number of algorithms agreed).

Usage

create_scoreSummary(container, classification_results)

Arguments

- container Class of type `matrix_container-class` generated by the `create_container` function.
- classification_results A `cbind()` of result objects returned by `classify_model`, or the object returned by `classify_models`.

Author(s)

Timothy P. Jurka <tpjurka@ucdavis.edu>, Loren Collingwood <lorenc2@uw.edu>

Examples

```r
library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data$["Title"], data$["Subject"]), language="english",
removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100,
virgin=FALSE)
models <- train_models(container, algorithms=c("RF", "SVM"))
results <- classify_models(container, models)
score_summary <- create_scoreSummary(container, results)
```

cross_validate

used for cross-validation of various algorithms.

Description

Performs n-fold cross-validation of specified algorithm.
Usage

cross_validate(container, nfold, algorithm = c("SVM", "SLDA", "BOOSTING", "BAGGING", "RF", "GLMNET", "TREE", "NNET"), seed = NA, method = "C-classification", cross = 0, cost = 100, kernel = "radial", maxitboost = 100, maxitglm = 10^5, size = 1, maxitnnet = 1000, MaxNWts = 10000, rang = 0.1, decay = 5e-04, ntree = 200, l1_regularizer = 0, l2_regularizer = 0, use_sgd = FALSE, set_heldout = 0, verbose = FALSE)

Arguments

container Class of type matrix_container-class generated by the create_container function.
nfold Number of folds to perform for cross-validation.
algorithm A string specifying which algorithm to use. Use print_algorithms to see a list of options.
seed Random seed number used to replicate cross-validation results.
method Method parameter for SVM implementation. See e1071 documentation for more details.
cross Cross parameter for SVM implementation. See e1071 documentation for more details.
cost Cost parameter for SVM implementation. See e1071 documentation for more details.
kernel Kernel parameter for SVM implementation. See e1071 documentation for more details.
maxitboost Maximum iterations parameter for boosting implementation. See caTools documentation for more details.
maxitglm Maximum iterations parameter for glmnet implementation. See glmnet documentation for more details.
size Size parameter for neural networks implementation. See nnet documentation for more details.
maxitnnet Maximum iterations for neural networks implementation. See nnet documentation for more details.
MaxNWts Maximum number of weights parameter for neural networks implementation. See nnet documentation for more details.
rang Range parameter for neural networks implementation. See nnet documentation for more details.
decay Decay parameter for neural networks implementation. See nnet documentation for more details.
ntree Number of trees parameter for RandomForests implementation. See randomForest documentation for more details.
l1_regularizer An numeric turning on L1 regularization and setting the regularization parameter. A value of 0 will disable L1 regularization. See maxent documentation for more details.
getStemLanguages

l2_regularizer: An numeric turning on L2 regularization and setting the regularization parameter. A value of 0 will disable L2 regularization. See maxent documentation for more details.

use_sgd: A logical indicating that SGD parameter estimation should be used. Defaults to FALSE. See maxent documentation for more details.

set_heldout: An integer specifying the number of documents to hold out. Sets a held-out subset of your data to test against and prevent overfitting. See maxent documentation for more details.

verbose: A logical specifying whether to provide descriptive output about the training process. Defaults to FALSE, or no output. See maxent documentation for more details.

Author(s)

Loren Collingwood, Timothy P. Jurka

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100,size=100,replace=FALSE),]
matrix <- create_matrix(cbind(data"Title","Subject"), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix,data$Topic.Code,trainSize=1:75, testSize=76:100, virgin=FALSE)
svm <- cross_validate(container,2,algorithm="SVM")

getStemLanguages

Description

This dynamically determines the names of the languages for which stemming is supported by this package. This is controlled when the package is created (not installed) by downloading the stemming algorithms for the different languages.

This language support requires more support for Unicode and more complex text than simple strings.

Usage

getStemLanguages()

Details

This queries the C code for the list of languages that were compiled when the package was installed which in turn is determined by the code that was included in the distributed package itself.
matrix_container-class

Value

A character vector giving the names of the languages.

Author(s)

Duncan Temple Lang <duncan@wald.ucdavis.edu>

References

See [http://snowball.tartarus.org/](http://snowball.tartarus.org/)

See Also

wordStem inst/scripts/download in the source of the Rstem package.

matrix_container-class

*an S4 class containing the training and classification matrices.*

Description

An S4 class containing all information necessary to train, classify, and generate analytics for a dataset.

Objects from the Class

Objects could in principle be created by calls of the form `new("matrix_container", ...)`. The preferred form is to have them created via a call to `create_container`.

Slots

- `training_matrix`: Object of class "matrix.csr": stores the training set of the DocumentTermMatrix created by `create_matrix`.
- `training_codes`: Object of class "factor": stores the training labels for each document in the `training_matrix` slot of `matrix_container-class`.
- `classification_matrix`: Object of class "matrix.csr": stores the classification set of the DocumentTermMatrix created by `create_matrix`.
- `testing_codes`: Object of class "factor": if `virgin=False`, stores the labels for each document in `classification_matrix`.
- `column_names`: Object of class "vector": stores the column names of the DocumentTermMatrix created by `create_matrix`.
- `virgin`: Object of class "logical": boolean specifying whether the classification set is virgin data (TRUE) or not (FALSE).

Author(s)

Timothy P. Jurka
Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100, virgin=FALSE)

container@training_matrix
container@training_codes
container@classification_matrix
container@testing_codes
container@column_names
container@virgin

NYTimes

a sample dataset containing labeled headlines from The New York Times.

Description

A sample dataset containing labeled headlines from The New York Times, compiled by Professor Amber E. Boydstun at the University of California, Davis.

Usage

data(NYTimes)

Format

A data.frame containing five columns.
2. Date - The date the headline appeared in The New York Times.
4. Subject - A manually classified subject of the headline.
5. Topic.Code - A manually labeled topic code corresponding to the subject.

Source

http://www.amberboydstun.com/

Examples

data(NYTimes)
read_data

print_algorithms

prints available algorithms for train_model() and train_models().

Description

An informative function that displays options for the algorithms parameter in train_model and train_models.

Usage

print_algorithms()

Value

Prints a list of available algorithms.

Author(s)

Timothy P. Jurka

Examples

library(RTextTools)
print_algorithms()

read_data

reads data from files into an R data frame.

Description

Reads data from several types of data storage types into an R data frame.

Usage

read_data(filepath, type=c("csv","delim","folder"), index=NULL, ...)

Arguments

filepath Character string of the name of the file or folder, include path if the file is not located in the working directory.

type Character vector specifying the file type. Options include csv, delim, and folder to denote .csv files, delimited files (tab, pipe, etc.) files, or folders of text files. If using the delim option, be sure to pass in a separate sep parameter to indicate how the file is delimited.
index  The path to a CSV file specifying the training label of each file in the folder of text files, one per line. An example of one line would be `1.txt,1`. Do not include the full file path for each file, that will be handled automatically using the folder location passed into `filepath`. This index file must be located outside the folder of files.

...  Other arguments passed to R’s `read.csv` function.

Value

An `data.frame` object is returned with the contents of the file.

Author(s)

Loren Collingwood, Timothy P. Jurka

Examples

```r
library(RTextTools)
data <- read_data(system.file("data/NYTimes.csv.gz", package="RTextTools"), type="csv", sep=";")
```

 recall_accuracy  calculates the recall accuracy of the classified data.

Description

Given the true labels to compare to the labels predicted by the algorithms, calculates the recall accuracy of each algorithm.

Usage

```r
recall_accuracy(true_labels, predicted_labels)
```

Arguments

- `true_labels`  A vector containing the true labels, or known values for each document in the classification set.
- `predicted_labels`  A vector containing the predicted labels, or classified values for each document in the classification set.

Author(s)

Loren Collingwood, Timothy P. Jurka
Examples

```r
library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"],data["Subject"][,]), language="english",
removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100,
virgin=FALSE)
models <- train_models(container, algorithms=c("RF","SVM"))
results <- classify_models(container, models)
analytics <- create_analytics(container, results)
recall_accuracy(analytics@document_summary$MANUAL_CODE,
analytics@document_summary$RF_LABEL)
recall_accuracy(analytics@document_summary$MANUAL_CODE,
analytics@document_summary$SVM_LABEL)
```

---

**summary.analytics** summarizes the **analytics-class** class

**Description**

Returns a summary of the contents within an object of class **analytics-class**.

**Usage**

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

**Arguments**

- **object**
  An object of class **analytics-class** containing the output of the `create_analytics` function.

- **...**
  Additional parameters to be passed onto the summary function.

**Author(s)**

Timothy P. Jurka
summary.analytics_virgin

summarizes the analytics_virgin-class class

Description

Returns a summary of the contents within an object of class analytics_virgin-class.

Usage

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

Arguments

object
An object of class analytics_virgin-class containing the output of the create_analytics function.

...
Additional parameters to be passed onto the summary function.

Author(s)

Timothy P. Jurka

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"])), language="english",
removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100,
virgin=TRUE)
models <- train_models(container, algorithms=c("RF","SVM"))
results <- classify_models(container, models)
analytics <- create_analytics(container, results)

summary(analytics)

train_model

makes a model object using the specified algorithm.

Description

Creates a trained model using the specified algorithm.
Usage

train_model(container, algorithm=c("SVM","SLDA","BOOSTING","BAGGING","RF","GLMNET","TREE","NNET"), method = "C-classification", cross = 0, cost = 100, kernel = "radial", maxitboost = 100, maxitglm = 10^5, size = 1, maxitnnet = 1000, MaxNWts = 10000, rang = 0.1, decay = 5e-04, trace=FALSE, ntree = 200, l1_regularizer = 0, l2_regularizer = 0, use_sgd = FALSE, set_heldout = 0, verbose = FALSE, ...

Arguments

container Class of type matrix_container-class generated by the create_container function.
algorithm Character vector (i.e. a string) specifying which algorithm to use. Use print_algorithms to see a list of options.
method Method parameter for SVM implementation. See e1071 documentation for more details.
cross Cross parameter for SVM implementation. See e1071 documentation for more details.
cost Cost parameter for SVM implementation. See e1071 documentation for more details.
kernel Kernel parameter for SVM implementation. See e1071 documentation for more details.
maxitboost Maximum iterations parameter for boosting implementation. See caTools documentation for more details.
maxitglm Maximum iterations parameter for glmnet implementation. See glmnet documentation for more details.
size Size parameter for neural networks implementation. See nnet documentation for more details.
maxitnnet Maximum iterations for neural networks implementation. See nnet documentation for more details.
MaxNWts Maximum number of weights parameter for neural networks implementation. See nnet documentation for more details.
rang Range parameter for neural networks implementation. See nnet documentation for more details.
decay Decay parameter for neural networks implementation. See nnet documentation for more details.
trace Trace parameter for neural networks implementation. See nnet documentation for more details.
ntree Number of trees parameter for RandomForests implementation. See randomForest documentation for more details.
l1_regularizer An numeric turning on L1 regularization and setting the regularization parameter. A value of 0 will disable L1 regularization. See maxent documentation for more details.
train_models

12_regularizer  An numeric turning on L2 regularization and setting the regularization parameter. A value of 0 will disable L2 regularization. See maxent documentation for more details.

use_sgd  A logical indicating that SGD parameter estimation should be used. Defaults to FALSE. See maxent documentation for more details.

set_heldout  An integer specifying the number of documents to hold out. Sets a held-out subset of your data to test against and prevent overfitting. See maxent documentation for more details.

verbose  A logical specifying whether to provide descriptive output about the training process. Defaults to FALSE, or no output. See maxent documentation for more details.

...  Additional arguments to be passed on to algorithm function calls.

Details

Only one algorithm may be selected for training. See train_models and classify_models to train and classify using multiple algorithms.

Value

Returns a trained model that can be subsequently used in classify_model to classify new data.

Author(s)

Timothy P. Jurka, Loren Collingwood

Examples

library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100, size=100, replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"], data["Subject"]), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix, data$Topic.Code, trainSize=1:75, testSize=76:100, virgin=FALSE)
rf_model <- train_model(container,"RF")
svm_model <- train_model(container,"SVM")

train_models  makes a model object using the specified algorithms.

Description

Creates a trained model using the specified algorithms.

Usage

train_models(container, algorithms, ...)
Arguments

container Class of type `matrix_container-class` generated by the `create_container` function.

algorithms List of algorithms as a character vector (e.g. c("SVM","MAXENT").

... Other parameters to be passed on to `train_model`.

Details

Calls the `train_model` function for each algorithm you list.

Value

Returns a list of trained models that can be subsequently used in `classify_models` to classify new data.

Author(s)

Wouter Van Atteveldt <wouter@vanatteveldt.com>

Examples

```r
library(RTextTools)
data(NYTimes)
data <- NYTimes[sample(1:3100,size=100,replace=FALSE),]
matrix <- create_matrix(cbind(data["Title"],data["Subject"])), language="english", removeNumbers=TRUE, stemWords=FALSE, weighting=tm::weightTfIdf)
container <- create_container(matrix,data$Topic.Code,trainSize=1:75, testSize=76:100, virgin=FALSE)
models <- train_models(container, algorithms=c("RF","SVM"))
```

Description

A sample dataset containing labeled bills from the United States Congress, compiled by Professor John D. Wilkerson at the University of Washington, Seattle and E. Scott Adler at the University of Colorado, Boulder.

Usage

data(USCongress)
wordStem

Format

A data.frame containing five columns.
1. ID - A unique identifier for the bill.
2. cong - The session of congress that the bill first appeared in.
3. billnum - The number of the bill as it appears in the congressional docket.
4. h_or_sen - A field specifying whether the bill was introduced in the House (HR) or the Senate (S).
5. major - A manually labeled topic code corresponding to the subject of the bill.

Source

http://www.congressionalbills.org/

Examples

data(USCongress)

---

wordStem

Get the common root/stem of words

Description

This function computes the stems of each of the given words in the vector. This reduces a word to its base component, making it easier to compare words like win, winning, winner. See http://snowball.tartarus.org/ for more information about the concept and algorithms for stemming.

Usage

wordStem(words, language = character(), warnTested = FALSE)

Arguments

words a character vector of words whose stems are to be computed.
language the name of a recognized language for the package. This should either be a single string which is an element in the vector returned by getStemLanguages, or alternatively a character vector of length 3 giving the names of the routines for creating and closing a Snowball SN\_env environment and performing the stem (in that order). See the example below.
warnTested an option to control whether a warning is issued about languages which have not been explicitly tested as part of the unit testing of the code. For the most part, one can ignore these warnings and so they are turned off. In the future, we might consider controlling this with a global option, but for now we suppress the warnings by default.
wordStem

Details

This uses Dr. Martin Porter’s stemming algorithm and the interface generated by Snowball [http://snowball.tartarus.org/](http://snowball.tartarus.org/).

Value

A character vector with as many elements as there are in the input vector with the corresponding elements being the stem of the word.

Author(s)

Duncan Temple Lang <duncan@wald.ucdavis.edu>

References

See [http://snowball.tartarus.org/](http://snowball.tartarus.org/)
Index

* IO
  getStemLanguages, 13
  wordStem, 23
* classes
  analytics-class, 2
  analytics_virgin-class, 3
  matrix_container-class, 14
* datasets
  NYTimes, 15
  USCongress, 22
* methods
  as.compressed.matrix, 3
  create_analytics, 2, 3, 6, 7, 8, 10–12, 14, 19
  create_container, 6, 7, 8, 10–12, 14, 20, 22
  create_ensembleSummary, 8
  create_matrix, 7, 8, 14
  create_precisionRecallSummary, 10
  create_scoreSummary, 11
  cross_validate, 11
  getStemLanguages, 13, 23
  matrix_container-class, 14
  NYTimes, 15
  print_algorithms, 12, 16, 20
  read_data, 16
  recall_accuracy, 17
  summary.analytics, 18
  summary.analytics_virgin, 19
  train_model, 4, 7, 16, 19, 22
  train_models, 4, 5, 7, 16, 21, 21
  USCongress, 22
  wordStem, 14, 23
* utilities
  getStemLanguages, 13
  wordStem, 23

analytics-class, 2, 18
analytics_virgin-class, 3, 19
as.compressed.matrix, 3

classify_model, 4, 5–7, 10, 11, 21
classify_models, 4, 5, 6, 7, 10, 11, 21, 22
create_analytics, 2, 3, 6, 7, 8, 18, 19