Description

This is the manual of the functions that were developed for the *FastTMtool*. The descriptions and details of the functions are formatted similarly to the official R packages. For more details, please review the official paper that introduces the first version of *FastTMtool*.

Accuracy_2_Vectors_new

Description: Classification evaluation metrics e.g. precision, recall, AUC, F1 score, accuracy as calculated by comparing two vectors. When the target variable is binary, values equal to 0 represent the negative class and values equal to 1 represent the positive class, where the values should be transformed into binary before calling the function. Otherwise, when the available classes of the test set are more than two. In this case, the evaluation metrics, apart from the accuracy, are calculated by setting every possible class as the positive class and by assessing the mean values of the calculated metrics as the final evaluation. For more details, please see the R package *MLmetrics*.

Arguments

predicts	Numerical or character vector representing the predictions usually produced from classification models.
test	Numerical or character vector representing the values of the test dataset (ground truth).

Value

A list object that contains the following components:

precision	The calculated precision evaluation measure. When the classes of the test dataset are more than two, this metric represents the average evaluation of the cases, where each possible class is set as positive.
recall	The calculated Recall evaluation measure. When the classes of the test dataset are more than two, this metric represents the average evaluation of the cases, where each possible class is set as positive.
f1	The calculated F1 score. When the classes of the test dataset are more than two, this metric represents the average evaluation of the cases, where each possible class is set as positive.
auc	The calculated AUC score. When the classes of the test dataset are more than two, this metric represents the average evaluation of the cases, where each possible class is set as positive.
accuracy	The calculated accuracy of the predictions.

auto_encoders

Description: Learning efficient data representations by compressing a numerical matrix into a new one. Often used for dimensionality reduction tasks.

Arguments

features	A numerical matrix of the initial data representations., where each row is an observation and each column is a feature.
Dimensions	The number of dimensions of the new data representations.

Value

A numerical matrix of the new data representations.

Document_vectors

Description: The main function for establishing document vectors via supervised learning. The function tensorflow_keras_nn_funs is called from this function as an expansion.

Arguments

word_vectors	A multidimensional matrix of word vectors included in the Document Term Matrix of item_list_text (rownames (word_vectors)= colnames (item_list_text$dtm). Useful only when type_words is set to dtm_ww.
item_list_text	A list object as returned from the function text_preprocessing
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
option	A character argument indicating whether to proceed with classification (nom_choice) or regression (con_choice) supervised learning. Default value is nom_choice. This argument matches the nom_con_var of the function tensorflow_keras_nn_funs.
type	A character argument indicating the supervised technique that will be employed. Available options: Starspace model (star_model), FastText (ft_model), Deep Averaging Network (dan_model), LSTM, RNN and CNN. Default value is star_model. It should be noted that the options star_model and ft_model are not available when the argument option is set to con_choice.
no_dims	The number of the desired dimensions of the extracted document vectors. Default value is 50.

type_words	A character value representing the selection of the words to be included in the training phase. Available options: Training with all words included in the texts (all_words), Training with the words included in the Document Term Matrix of item_list_text with (dtm_ww) or without (dtm_nw) initialized weights. In case of initialized word vectors the user must provide a numerical matrix (word_vectors). Default value is all_words.

Value

A numerical matrix representing the extracted document vectors.

fclust_mapping_with_npmi

Description: A function for fitting three different word clustering approaches for topic extraction. These approaches are based on the Fuzzy k-means technique, the Gaussian Mixture Model based clustering and Leiden algorithm for network clustering and community detection. The first two approaches are based on the topology of word vectors while the third approach is based on similarity-dissimilarity measures extracted from either word vectors or the inclusion index. For more, details please read the paper of this tool.

Arguments

word_vectors	A multidimensional matrix of word vectors included in the Document Term Matrix tSparse_train. (rownames (word_vectors)= colnames (tSparse_train).
min_topics	Minimum number of topics to be evaluated. Only used when the argument type is not set to leiden. The default value is 2.
topic_range	Maximum number of topics to be evaluated. Only used when the argument type is not set to leiden. The default value is 20.
tSparse_train	A Document Term Matrix.
center_top_Words	A boolean value indicating whether the top words would be evaluated based on their frequencies or not. Values equal to FALSE indicate that the word frequencies and cluster memberships would be used to identify the top words of each cluster. Otherwise only the cluster memberships are evaluated. Default values is FALSE.
nn	Number of neighbors to be used for the UMAP dimensionality reduction. Only useful when dim_red_options is equal to umap_red. Default values is 5.
l	Number of terms to be used in the evaluation of the topic coherence of each model. Default value is 10.
spr	Spread parameter of the UMAP algorithms. Only useful when dim_red_options is equal to umap_red. Default value is 1.

md	Parameter representing the minimum distance of the nearest neighbors in a projection extracted from UMAP. Only useful when dim_red_options is equal to umap_red. Default value is 0.01.
type	Character value indicating which clustering approach will be employed. Available options: Fuzzy k-means (fclust), Gaussian Mixture Model based clustering (mclust) and Leiden algorithm for network clustering and community detection (leiden). Default value is fclust.
umap_metric	Character value for the distance metric to be used for the UMAP algorithm. Only useful when dim_red_options is equal to umap_red. For more options please read the documentation of the R package uwot.
glove_leiden	A Boolean value indicating whether to proceed with cosine similarity measures from word vectors or with similarity measures evaluated from the Term Co-occurrence Matrix, from item_list_text, for the leiden algorithm. Only useful when the argument type is set to leiden. Default value is FALSE.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.
ii_rev	A boolean value indicating whether to proceed with the reverse inclusion index similarity measure or not. Only useful when the argument type is set to leiden and the argumnet glove_leiden is set to FALSE. Default value is TRUE.
leiden_option_mem	Currently not supported, should be always set to 2. Default value is 2.
no_clust_mem_cond	Boolean argument indicating whether the unconnected nodes of a graph should be considered as a separate cluster. Only used when type is set to leiden. Default value is FALSE.
no_umap_dims	Number of dimensions of the vectors produced from a dimensionality reduction technique. Useful when dim_red_options is not set to no_red. Default value is 2.
dim_red_options	Which type of dimensionality reduction should be applied to the initialized word vectors. Available options: Uniform Manifold Approximation and Projection for Dimension Reduction (umap_red), Principal Component Analysis (pca_red), Singular Value Decomposition (svd_red), t-distributed stochastic neighbor

	embedding (tsne_red), factor analysis (factanal_red). If this option is set to no_red then initialized word vectors are not preprocessed. Default value is no_red.
stand_leiden_words_mem	Boolean value indicating whether the memberships extracted from the word clustering approach that is based on the leiden algorithm should be standardized. Default value is FALSE.

Value

A list of different objects depending on the selected approach:

phi	A matrix representing the distributions of words (columns) over topics (rows). Each row sums to one.
short_visualization	A visualization of the clusters extracted from each algorithm. If the word vectors are not initial or reduced to a 2d matrix for the fclust and mclust options (argument type) then the Principal Component Analysis is employed to produce an effective visualization of the clusters. In this case only the top words are projected for the two aforementioned options.
full_visualization	A visualization of the clusters extracted from each algorithm. If the word vectors are not initial or reduced to a 2d matrix for the fclust and mclust options (argument type) then the Principal Component Analysis is employed to produce an effective visualization of the clusters. In this case all the available words of the Document Term Matrix are projected. This feature is not available when the argument type is set to leiden.
document_memberships	A matrix representing the produced distributions of topics (columns) over the investigated documents (rows), usually referred as theta.
coherence_npmi	A list object that stores the topic coherence of all the evaluated models based on the predefined options.
max_coh	The topic coherence of the final models. Only the model that produces the highest evaluation is selected.
top_terms	A matrix that contains top words of each topic that were also used to calculate the topic coherence of the extracted model. Each column represents the top words of a topic.
topic_vis	Two-dimensional interactive topic model visualization using the R package LDAvis.
f_clust	Object as returned from the FKM function of the R package *fclust*
m_clust	Object as returned from the Mclust function of the R package *mclust*
leiden_clust	Object as returned from the cluster_leiden function of the R package *igraph*

Feature_evaluation_methods

Description: Feature evaluation based on filtering techniques. Currently, several of the filtering techniques included in the R package *praznik* are supported along with cosine similarity and the spearman correlation coefficient. Currently, the feature representations of Document Term Matrix and Dichotomized Document Term Matrix are supported, using the information included in the item_list_text.

Arguments

item_list_text	A list object as returned from the function text_preprocessing
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.
method_feature	A character variable representing the filtering technique that will be employed for feature evaluation. The available options are the following: Minimal joint mutual information maximization filter (jmim_ff), Mutual Information Maximization filter (mim_ff), Minimum redundancy maximal relevancy filter (mrmr_ff), Joint Mutual Information (jmi_ff), Double input symmetrical relevance filter (disr_ff), Minimal normalised joint mutual information maximisation filter (njmim_ff), Minimal conditional mutual information maximisation filter (cmim_ff), Joint impurity filter (jim_ff), Conditional mutual information maximisation filter (cmi_ff), cosine similarity (cossimil_ff) and sperman correlation coefficient (spearman_ff).
matrix_feature	A character value indicating the document representation to be used for feature evaluation. Currently the Document Term Matrix (dtm_mf) and the Dichotomized Document Term Matrix is supported (dtmd_mf)
no_feature	The number of the most highly evaluated features to be returned.

Value

A data frame that contains the names of the returned features in the first column and their evaluation in the second column.

find_coh

Description: A function that calculates the Normalized Pointwise Mutual Information (NPMI) given a the top words of a model and the term co-occurrence matrix, usually as returned by the function text_preprocessing.

Arguments

ldaOut.terms	The top terms of each topic, where each column represents the top terms of a topic.
tcm	The Term Co-occurrence Matrix in a similar format with the one produced by the function text_preprocessing.
rows_train	The number of observations included in the training dataset (rows of the training dataset).

Value

Provides a singular value that represents the mean coherence of the topics as evaluated from the top terms of each topic.

prepare_glove

Description: Produces word vectors based on the GloVe algorithm.

Arguments

item_list_text	A list object as returned from the function text_preprocessing
glove_skipgram_clause	Boolean value indicating whether to calculate the similarities between the words using the skipgram architecture. Default value is TRUE.
ws	Integer value representing the window size used on the skipgram architecture.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
full_tcm_clause	When the argument glove_skipgram_clause is FALSE, the user can select whether to calculate the full word co-occurrences or the binary co-occurrences as inputs in the GloVe. Values equal to FALSE indicate that the word similarities will be calculated based on binary co-occurrences. The default value is FALSE.
dimensions	The number of the desired dimensions of the extracted word vectors. Default value is 200.

Value

A numerical matrix representing the extracted word vectors. The row names of the extracted vectors match the words included in the Document Term Matrix of the item_list_text list.

tensorflow_keras_nn_funs

Description: This function provides options of three architectures of neural networks, e.g. Long short-term memory, Recurrent Neural Networks and Convolutional Neural Networks for supervised learning in order to produce document vectors.

Arguments

all_set_text_final	The initial text to be processed.
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
type	A character argument indicating the architecture of neural network to be employed. Available options: LSTM, RNN, CNN. Default value is LSTM.
nom_con_var	A character argument indicating whether to proceed with classification (nom_choice) or regression (con_choice) supervised learning. Default value is nom_choice.
imbalance_cond	Boolean value representing a balancing option, only in case of a classification task e.g. when nom_con_var is set to nom_choice. When this argument is TRUE the minority classes are given higher weights than the majority class during the training phase. Default value is TRUE.
no_dims	The number of the desired dimensions of the extracted document vectors. Default value is 50.
type_words	A character value representing the selection of the words to be included in the training phase. Available options: Training with all words included in the texts (all_words), Training with the words included in the Document Term Matrix of item_list_text with (dtm_ww) or without (dtm_nw) initialized weights. In case of initialized word vectors the user must provide a numerical matrix (word_vectors). Default value is all_words.
item_list_text	A list object as returned from the function text_preprocessing

word_vectors	A multidimensional matrix of word vectors included in the Document Term Matrix of item_list_text (rownames (word_vectors)= colnames (item_list_text$dtm). Useful only when type_words is set to dtm_ww.

Value

A numerical matrix representing the extracted document vectors.

text_preprocessing

Description: Provides several approaches for Natural Language Processing including text preprocessing, ngrams construction and Document Term Matrices. A user can proceed by applying only lower case transformation and punctuation removal (basic_preprocess) or investigate several options on text preprocessing. When the user selects the second approach, word elongation and contraction replacement is always applied apart from the rest options that are selected.

Arguments

all_set_text	The initial text to be processed.
ngrams_clause	Boolean value indicating whether to construct ngrams or not. Default value is FALSE.
min_doc_r	Minimum proportion of documents a word should occur to be included in the constructed Document Term Matrix. Default value is 0.002.
max_doc_r	Maximum proportion of documents a word should occur to be included in the constructed Document Term Matrix. Default value is 0.5.
ret_dtm	Boolean value indicating whether to construct a Document Term Matrix or not. In order to be able to use the majority of the available functions of this package, this value should be set to TRUE. Default value is TRUE.
do_stem	Apply stemming to the tokens of the documents. Default value is FALSE.
do_rmv_stop	Remove stop words. Default value is TRUE.
do_lower_case	Apply lower case transformation. Default value is TRUE.
do_rmv_mention	Remove mentions e.g. remove strings that start with the character @. Default value is TRUE.
do_rpl_number	Replace a numerical token into character e.g. the token 103 will be transformed into one hundred and three while the token file_3 will remain file_3. Default value is TRUE.

do_rpl_hash	Replace hashtags e.g. remove strings that start with the character #. Default value is TRUE.
do_rpl_html	Replaces HTML markup. Default value is TRUE
do_rpl_qmark	Replace question marks with the token questionmark. Default value is TRUE.
do_rpl_emark	Replace exclamation marks with the token exclamationmark. Default value is TRUE.
do_rpl_punct	Replace punctuation. Default value is TRUE.
do_rpl_digit	Replace digits in every token. For example the token 1Jo5e23 will be transformed into Jo e. Default Value is TRUE.
basic_preprocess	Boolen value indicating whether to perform only bacis preprocessing, e.g. lower case transformation and punctuation removal, or not. Apart from the arguments do_rmv_stop and do_stem, every other preprocessing option is ignored. In this case word elongation and contraction replacement are not applied. Usually, word2vec and doc2vec algorithms use this type of preprocessing. Default value is TRUE.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
is_tfidf	Boolean argument that indicates whether to calculate Term Frequency – Inverse Document Frequency or just Term Frequency as the word weightings of the Document Term Matrix. When this argument is set to FALSE, the Term Frequency weighting is calculated. Default value is FALSE.
min_ngrams	Minimum length of a ngram. Works only when ngrams_clause is TRUE. Default value is 2.
max_ngrams	Maximum length of a ngram. Works only when ngrams_clause is TRUE. Default value is 4.

Value

Returns a list object that is referred as item_list_text in the whole package containing the following components:

old_words	Original terms, character vector, included in the processed text and belong to the dtm. This step is necessary as some tokens may cause problems to several R functions. For example, the word function refers to a constructed function and should be transformed into

	"function.". Also, numerical tokens could cause similar problems, as a result the number 2023 is also transformed into the token X2023. In our case, tokens of this type are transformed and stored into the dtm while the original ones are stored into this vector.
text	The extracted text as produced after the preprocessing steps.
dtm	The constructed Document Term Matrix.
tcm	The Term Co-occurrence Matrix that stores information regarding the number of documents that two words co-occur. The diagonal values indicate the number of documents each word occur.

topic_models

Description: Performs topic modelling algorithms based on various existing R packages

Arguments

item_list_text	A list object as returned from the function text_preprocessing
word_vectors	A multidimensional matrix of word vectors included in the Document Term Matrix of item_list_text (rownames (word_vectors)= colnames (item_list_text$dtm). Useful only when ETM algorithm is employed.
type	A string argument that specifies which algorithm will be used. Possible values: LDA_vem, CTM_vem, STM_vem, ETM, LSA, LDA_m. The default value is LDA_vem.
no_topics	Number of topics to be evaluated. The default value is 10.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
alpha_var	Symmetric value of the alpha prior parameter of LDA_m. The default value is 1.
beta_var	Symmetric value of the beta prior parameter of LDA_m. The default value is 1.
iter_var	Number of iterations of the LDA_m algorithm. The default values is 10
as_alpha	Boolean value indicating whether to set asymmetric alpha values or not. When this parameter is TRUE, the value of alpha_var is not used and the i-th topic is given the following alpha value: $\frac{1}{i+\sqrt{no\_topics}}$. The default value is FALSE.

no_top_terms	Number of terms to be used in the evaluation of the topic coherence of each model. Default value is 10.
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.

Value

Returns a list object with the following components:

phi	A matrix representing the distributions of words (columns) over topics (rows). Each row sums to one.
model	The final model that is constructed based on the predefined arguments. It may be used for future predictions or re training.
keyword_table	A matrix that contains top words of each topic that were also used to calculate the topic coherence of the extracted model. Each column represents the top words of a topic.
coherence_npmi	The mean topic coherence of the extracted model based on the Normalized Pointwise Mutual Information (NPMI).
document_memberships	A matrix representing the produced distributions of topics (columns) over the investigated documents (rows), usually referred as theta.
topic_vis	Two-dimensional interactive topic model visualization using the R package LDAvis. Not available for the type option LSA.

Train_Regression

Description: Training of machine learning regression models using the observations of the train dataset. The test dataset is used to evaluate the constructed models. Predictions and evaluation metrics are produced for each model. Currently implementations of Generalized Linear Models, Gradient Boosting Machines, Random Forest and Deep Learning are supported. An ensemble model that predicts the median of the predictions produced by the previous models, for each observation, is also evaluated.

Arguments

features	A numerical matrix of document vectors that constitute the inputs-features of the documents.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.

Value

Returns a list object that contains the predictions and evaluation metric of each classification model:

eval_list	A numerical matrix representing the performance evaluation of each model. The following evaluation metrics are currently supported: Precision, Recall, F1 score, AUC and Accuracy.
pred_list	A list object that stores the predictions of each constructed model.

train_test_functions

Description: Training of machine learning classification models using the observations of the train dataset. The test dataset is used to evaluate the constructed models. Predictions and evaluation metrics are produced for each model. Currently implementations of Generalized Linear Models, Gradient Boosting Machines, Naïve Bayes, Random Forest and Deep Learning are supported. An ensemble model that predicts the median of the predictions produced by the previous models, for each observation, is also evaluated.

Arguments

features	A numerical matrix of document vectors that constitute the inputs-features of the documents.
split2	A Boolean vector that indicates which observations (index) belong to the training and test dataset. TRUE values correspond to the train dataset while FALSE values correspond to the test dataset. The observations that belong to the train dataset are used for every training procedure, including the construction of Document Term Matrices, while the test dataset is ignored.
categories_assignement	Numerical vector (float or integer) that contains the values of the target variable of each observation.
imbalance_cond	Boolean value representing a balancing option. When this argument is TRUE the minority classes are given higher weights than the majority class during the training phase. Default value is TRUE.
weight_or_balance	Currently not supported, should be also set to weight_choice. Default value is weight_choice.

Value

Returns a list object that contains the predictions and evaluation metric of each classification model:

eval_list	A numerical matrix representing the performance evaluation of each model. The following evaluation metrics are currently supported: Precision, Recall, F1 score, AUC and Accuracy.
pred_list	A list object that stores the predictions of each constructed model.