Package 'LinkOrgs'

December 14, 2025

Title LinkOrgs: Algorithms for Organizational Record Linkage

Version 0.01

Description An R package for organizational records using the algorithms of Jerzak & Libgober (2023+). The linkage is done based on organizational names and using half a billion open collaborated records on those names from LinkedIn users. It also contains functions implementing string matching performance metrics, as well as a fast, parallized version of fuzzy string matching.

Depends R (>= 3.3.3)

License
Creative Commons Attribution-Noncommercial-No Derivative Works 4.0, for academic use only.

Encoding UTF-8

LazyData true

Maintainer 'Connor Jerzak' <connor.jerzak@gmail.com>

Imports
data.table,plyr,Rfast,stringdist,parallel,glmnet,stringr,dplyr,fastmatch,reticulate,R.utils,foreach,doParallel

RoxygenNote 7.3.3

RemoteType local

RemotePkgRef local::~/Documents/LinkOrgs-software/LinkOrgs

RemoteUrl /Users/cjerzak/Documents/LinkOrgs-software/LinkOrgs

Contents

AssessMatchPerformance

AssessMatchPerformance

Description

Automatically computes the true/false positive and true/false negative rates based on a ground-truth (preferably human-generated) matched dataset.

Usage

```
AssessMatchPerformance(x,y,by,...)
```

Arguments

x, y	data frames to be merged
z	the merged data frame to be analyzed. Should contain by,by.x, and/or by.y as column names, depending on usage.
z_true	a reference data frame containing target/true matched dataset. Should contain by,by.x, and/or by.y as column names, depending on usage.
by, by.x, by.y	character strings specifying of the columns used for merging.

Value

ResultsMatrix A matrix containing the information on the true positive, false positive, true negative, and false negative rate, in addition to the matched dataset size. These quantities are calculated based off all possible nrow(x)*nrow(y) match pairs.

Examples

```
# Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)
z <- data.frame("orgnames_x"=x_orgnames[1:2], "orgnames_y"=y_orgnames[1:2])
z_true <- data.frame("orgnames_x"=x_orgnames, "orgnames_y"=y_orgnames)

# Obtain match performance data
PerformanceMatrix <- AssessMatchPerformance(x = x,
                                    y = y,
                                    z = z,
                                    z_true = z_true,
                                    by.x = "orgnames_x",
                                    by.y = "orgnames_y")
print( PerformanceMatrix )
```

`BuildBackend` *Build the environment for LinkOrgs machine learning models. Builds a conda environment in which jax, optax, equinox, and jmp are installed.*

Description

Build the environment for LinkOrgs machine learning models. Builds a conda environment in which jax, optax, equinox, and jmp are installed.

Usage

```
BuildBackend(conda_env = "LinkOrgs_env", conda = "auto", tryMetal = T)
```

Arguments

`conda_env`	(default = `"LinkOrgs_env"`) Name of the conda environment in which to place the backends.
`conda`	(default = `auto`) The path to a conda executable. Using `"auto"` allows reticulate to attempt to automatically find an appropriate conda binary.

Value

Builds the computational environment for `LinkOrgs`. This function requires an Internet connection. You may find out a list of conda Python paths via: `system("which python")`

Examples

```
# For a tutorial, see
# github.com/cjerzak/linkorgs-software/
```

`dropboxURL2downloadURL` *dropboxURL2downloadURL*

Description

Downloads

Usage

```
dropboxURL2downloadURL(url)
```

Arguments

`url`	character string with the URL housing the data object.
`target_extension`	(default = `".csv"`) character string describing the target extension of the file in the downloaded .zip folder.

Details

dropboxURL2downloadURL

Value

z The

Examples

```
# Example download
my_dt <- dropboxURL2downloadURL(url="https://www.dropbox.com/s/iqf9ids77dckopf/Directory_LinkIt_bipartite_E
```

GetCalibratedDistThres

GetCalibratedDistThres

Description

Calibrates a distance threshold based on a target average number of matches per alias. Samples pairwise distances from a subset of observations to determine the threshold that would yield approximately the desired number of matches.

Usage

```
GetCalibratedDistThres(
  x = NULL,
  by.x = NULL,
  y = NULL,
  by.y = NULL,
  AveMatchNumberPerAlias = 5L,
  qgram = 2L,
  DistanceMeasure = "jaccard",
  nCores = NULL,
  mode = "euclidean"
)
```

Arguments

x	Input data. For mode = "euclidean": an embedding matrix where rows are observations and columns are embedding dimensions. For mode = "discrete": a data frame containing the column specified by by.x.
by.x	Column name in x to use for matching. Only used when mode = "discrete".
y	Input data. For mode = "euclidean": an embedding matrix where rows are observations and columns are embedding dimensions. For mode = "discrete": a data frame containing the column specified by by.y.
by.y	Column name in y to use for matching. Only used when mode = "discrete".
AveMatchNumberPerAlias	Target average number of matches per observation. Used to calibrate the distance threshold. Default is 5.

qgram	The q-gram size for string distance calculation. Only used when mode = "discrete". Default is 2.
DistanceMeasure	The string distance measure to use. Only used when mode = "discrete". Options include "jaccard", "osa", "jw". See ?stringdist::stringdist for all options. Default is "jaccard".
nCores	Number of CPU cores for parallel computation. Only used when mode = "discrete". Default is NULL (auto-detect).
mode	Character string specifying the distance computation mode. Must be either "euclidean" (for embedding-based matching) or "discrete" (for string-based matching). Default is "euclidean".

Value

A numeric value representing the calibrated distance threshold.

LinkOrgs *LinkOrgs*

Description

Implements the organizational record linkage algorithms of Libgober and Jerzak (2023+) using half-a-billion open-collaborated records.

Usage

```
LinkOrgs(
  x = NULL,
  y = NULL,
  by = NULL,
  by.x = NULL,
  by.y = NULL,
  embedx = NULL,
  embedy = NULL,
  embedDistMetric = NULL,
  algorithm = "ml",
  conda_env = "LinkOrgs_env",
  conda_env_required = T,
  ReturnDiagnostics = F,
  ReturnProgress = T,
  ToLower = T,
  NormalizeSpaces = T,
  RemovePunctuation = T,
  MaxDist = NULL,
  MaxDist_network = NULL,
  AveMatchNumberPerAlias = 10,
  AveMatchNumberPerAlias_network = 2,
  DistanceMeasure = "jaccard",
  qgram = 2,
  RelThresNetwork = 1.5,
```

```
  ml_version = "v1",
  openBrowser = F,
  ExportEmbeddingsOnly = FALSE,
  ReturnDecomposition = FALSE,
  python_executable,
  nCores = NULL,
  deezyLoc = NULL
)
```

Arguments

x, y	data frames to be merged
by, by.x, by.y	character vector(s) that specify the column names used for merging data frames x and y. The merging variables should be organizational names. See ?base::merge for more details regarding syntax.
algorithm	character; specifies which algorithm described in Libgober and Jerzak (2023+) should be used. Options are "markov", "bipartite", "ml", and "transfer". Default is "ml", which uses a machine-learning approach using Transformer netes and 9 million parameters to predict match probabilities using half a billion open-collaborated recoreds as training data.
conda_env	character string; specifies a conda environment where JAX and related packages have been installed (see ?LinkOrgs::BuildBackend). Used only when algorithm='ml' or DistanceMeasure='ml'.
conda_env_required	Boolean; specifies whether conda environment is required.
ReturnDiagnostics	Boolean; specifies whether various match-level diagnostics should be returned in the merged data frame.
ml_version	character; specifies which version of the ML algorithm should be used. Options are of the form "v0" and "v1". Highest version currently supported is "v1" (11M parameters).
ExportEmbeddingsOnly	Boolean; if TRUE with algorithm='ml' (or DistanceMeasure='ml'), return only ML embeddings for x and/or y without matching for offline linkage.
...	For additional specification options, see "Details".

Details

LinkOrgs automatically processes the name text for each dataset (specified by by or by.x, and by.y. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. Default is "jaccard". To use the combined machine learning and network methods, set algorithm to "bipartite" or "markov", and DistanceMeasure to "ml".
- Set MaxDist to control the maximum allowed distance between two matched strings
- Set MaxDist_network to control the maximum allowed distance between two matched strings in the integration with the LinkedIn network representation.
- Set AveMatchNumberPerAlias to control the maximum allowed distance between two matched strings. Takes priority over MaxDist if both specified.

- Set AveMatchNumberPerAlias_network to control the maximum allowed distance between two matched strings in the integration with the LinkedIn network representation. Takes priority over MaxDist_network if both specified.
- Set qgram to control the character-level q-grams used in the distance measure. Default is 2.
- Set RemoveCommonWords to TRUE to remove common words (those appearing in > 10% of aliases). Default is FALSE.
- Set NormalizeSpaces to TRUE to remove hanging whitespaces. Default is TRUE.
- Set RemovePunctuation to TRUE to remove punctuation. Default is TRUE.
- Set ToLower to TRUE to ignore case. Default is TRUE.

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs <- LinkOrgs(x = x,
                       y = y,
                       by.x = "orgnames_x",
                       by.y = "orgnames_y",
                       MaxDist = 0.6)

print( linkedOrgs )
```

---

pDistMatch_discrete *pFuzzyMatch_discrete*

Description

Performs parallelized fuzzy matching of strings based on the string distance measure specified in DistanceMeasure. Matching is parallelized using all available CPU cores to increase execution speed.

Usage

```
pDistMatch_discrete(
  x,
  y,
  by = NULL,
  by.x = NULL,
  by.y = NULL,
  embedDistMetric = NULL,
  return_stringdist = T,
```

```
  onlyUFT = T,
  qgram = 2,
  DistanceMeasure = "jaccard",
  MaxDist = 0.2,
  ReturnProgress = T,
  nCores = NULL,
  ReturnMaxDistThreshold = F
)
```

Arguments

x, y	data frames to be merged
by, by.x, by.y	specifications of the columns used for merging. We follow the general syntax of base::merge; see ?base::merge for more details.
...	For additional options, see "Details".

Details

pFuzzyMatch can automatically process the by text for each dataset. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")
- Set MaxDist to control the maximum allowed distance between two matched strings
- Set AveMatchNumberPerAlias to control the maximum allowed distance between two matched strings. Takes priority over MaxDist if both specified.
- Set qgram to control the character-level q-grams used in the distance measure. (Default is 2)
- Set RemoveCommonWords to TRUE to remove common words (those appearing in > 10% of aliases). (Default is FALSE)
- Set NormalizeSpaces to TRUE to remove hanging whitespaces. (Default is TRUE)
- Set RemovePunctuation to TRUE to remove punctuation. (Default is TRUE)
- Set ToLower to TRUE to ignore case. (Default is TRUE)

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)
z <- data.frame("orgnames_x"=x_orgnames[1:2], "orgnames_y"=y_orgnames[1:2])
z_true <- data.frame("orgnames_x"=x_orgnames, "orgnames_y"=y_orgnames)

# Perform merge
linkedOrgs_fuzzy <- pFuzzyMatch(x = x,
                      y = y,
                      by.x = "orgnames_x",
```

```
                        by.y = "orgnames_y")
```

---

pDistMatch_euclidean *pDistMatch_euclidean*

Description

Performs parallelized distance computation strings based on the string distance measure specified in DistanceMeasure. Matching is parallelized using all available CPU cores to increase execution speed.

Usage

```
pDistMatch_euclidean(
  embedx,
  embedy,
  MaxDist = NULL,
  embedDistMetric = NULL,
  ReturnProgress = T
)
```

Arguments

x, y	data frames to be merged
by, by.x, by.y	specifications of the columns used for merging. We follow the general syntax of base::merge; see ?base::merge for more details.
...	For additional options, see "Details".

Details

pDistMatch_euclidean can automatically process the by text for each dataset. Users may specify the following options:

- Set DistanceMeasure to control algorithm for computing pairwise string distances. Options include "osa", "jaccard", "jw". See ?stringdist::stringdist for all options. (Default is "jaccard")

Value

z The merged data frame.

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)
z <- data.frame("orgnames_x"=x_orgnames[1:2], "orgnames_y"=y_orgnames[1:2])
z_true <- data.frame("orgnames_x"=x_orgnames, "orgnames_y"=y_orgnames)
```

```
# Perform merge
linkedOrgs_fuzzy <- pFuzzyMatch(x = x,
                        y = y,
                        by.x = "orgnames_x",
                        by.y = "orgnames_y")
```

---

pFuzzyMatch_discrete *pFuzzyMatch_discrete*

Description

Implements

Usage

```
pFuzzyMatch_discrete(
  x = NULL,
  by.x = NULL,
  embedx = NULL,
  y = NULL,
  by.y = NULL,
  embedy = NULL,
  embedDistMetric = NULL,
  MaxDist = NULL,
  qgram = 2,
  DistanceMeasure = "jaccard",
  AveMatchNumberPerAlias = NULL,
  nCores = NULL,
  ...
)
```

Arguments

x, y	data frames to be merged

Details

...

Value

...

Examples

```
#Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)
```

pFuzzyMatch_euclidean *pFuzzyMatch_euclidean*

Description

Implements

Usage

```
pFuzzyMatch_euclidean(
  x = NULL,
  by.x = NULL,
  embedx = NULL,
  y = NULL,
  by.y = NULL,
  embedy = NULL,
  embedDistMetric = NULL,
  MaxDist = NULL,
  AveMatchNumberPerAlias = NULL,
  ...
)
```

Arguments

x, y	data frames to be merged

Details

...

Value

...

Examples

```
#' #Create synthetic data
x_orgnames <- c("apple","oracle","enron inc.","mcdonalds corporation")
y_orgnames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orgnames_x"=x_orgnames)
y <- data.frame("orgnames_y"=y_orgnames)
```

print2 *print2*

Description

Prints a message with a timestamp prefix.

Usage

```
print2(text, quiet = F)
```

Arguments

text	Character string to print.
quiet	Logical; if TRUE, suppress output. Default is FALSE.

Value

Invisibly returns NULL. Called for its side effect of printing.

Examples

```
print2("Hello world!")
```

url2dt *url2dt*

Description

Downloads a .zip file from a URL as a data.table from a URL.

Usage

```
url2dt(url)
```

Arguments

url	character string with the URL housing the data object.
target_extension	(default = ".csv") character string describing the target extension of the file in the downloaded .zip folder.

Details

url2dt downloads a zipped .csv file and loads it into memory based on the input URL.

Value

z The downloaded data object from the URL.

Examples

```
# Example download
my_dt <- url2dt(url="https://www.dropbox.com/s/iqf9ids77dckopf/Directory_LinkIt_bipartite_Embeddings.csv.zi
```

Index