

# Package ‘LinkOrgs’

February 22, 2023

**Title** LinkOrgs: Algorithms for Organizational Record Linkage

**Version** 0.0

**Authors** 'Brian Libgober <libgober@gmail.com> [aut,cre], Connor Jerzak <connor.jerzak@gmail.com> [aut, cre]'

**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,doMC,parallel,glmnet,parallel,stringr,dplyr,fastmatch,reticulate

**RoxygenNote** 7.2.1

## R topics documented:

AssessMatchPerformance . . . . .	2
BuildML . . . . .	3
FastFuzzyMatch . . . . .	3
LinkOrgs . . . . .	4
RestoreML . . . . .	6
TrainML . . . . .	6
<b>Index</b>	7

---

AssessMatchPerformance

*AssessMatchPerformance*


---

## Description

Computes the true/false positive and true/false negative rates of a candidate matching based on a ground-truth (preferably human-generated) matched dataset.

## Usage

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

## Arguments

<code>x, y</code>	data frames to be merged
<code>z</code>	the merged data frame to be analyzed. Should contain <code>by</code> , <code>by.x</code> , and/or <code>by.y</code> as column names, depending on usage.
<code>z_true</code>	a reference data frame containing target/true matched dataset. Should contain <code>by</code> , <code>by.x</code> , and/or <code>by.y</code> as column names, depending on usage.
<code>by, by.x, by.y</code>	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)` candidate match pairs.

## Examples

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

# Obtain match performance data
PerformanceMatrix <- AssessMatchPerformance(x = x,
                                             y = y,
                                             z = z,
                                             z_true = z_true,
                                             by.x = "orenames_x",
                                             by.y = "orenames_y")

print( PerformanceMatrix )
```

---

BuildML	<i>A primarily internal function which builds the organizational record linkage models used in Libgober and Jerzak (2023+).</i>
---------	---

---

**Description**

A primarily internal function which builds the organizational record linkage models used in Libgober and Jerzak (2023+).

**Usage**

```
BuildML(...)
```

---

FastFuzzyMatch	<i>FastFuzzyMatch</i>
----------------	-----------------------

---

**Description**

Performs fast 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**

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

**Arguments**

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

**Details**

FastFuzzyMatch 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_orenames <- c("apple","oracle","enron inc.,"mcdonalds corporation")
y_orenames <- c("apple corp","oracle inc","enron","mcdonalds co")
x <- data.frame("orenames_x"=x_orenames)
y <- data.frame("orenames_y"=y_orenames)
z <- data.frame("orenames_x"=x_orenames[1:2], "orenames_y"=y_orenames[1:2])
z_true <- data.frame("orenames_x"=x_orenames, "orenames_y"=y_orenames)

# Perform merge
linkedOrgs_fuzzy <- FastFuzzyMatch(x = x,
                                   y = y,
                                   by.x = "orenames_x",
                                   by.y = "orenames_y")
```

---

LinkOrgs	<i>LinkOrgs</i>
----------	-----------------

---

Description

Implements the organizational record linkage algorithms of Libgober and Jerzak (2023+).

Usage

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

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", and "ml". Default is "ml", which uses a machine learning approach to predicting the match probability.
- conda\_env             character string; specifies a conda environment where tensorflow and related packages have been installed. Used only when algorithm='ml' or DistanceMeasure='ml'.
- ReturnDiagnostics     logical; specifies whether various match-level diagnostics should be returned in the merged data frame.
- ...                    For additional specification options, see "Details".



---

`RestoreML`*RestoreML*

---

**Description**

A function, primarily for internal used, used to initialize the machine learning models used in the record linkage algorithms of Libgober and Jerzak.

**Usage**

```
RestoreML(...)
```

---

`TrainML`*TrainML*

---

**Description**

Internal function that performs the training of the machine learning models used for organizational record linkage algorithms of Libgober and Jerzak.

**Usage**

```
TrainML(...)
```

# Index

AssessMatchPerformance, [2](#)

BuildML, [3](#)

FastFuzzyMatch, [3](#)

LinkOrgs, [4](#)

RestoreML, [6](#)

TrainML, [6](#)