Package 'dialectR'

May 20, 2021

Type Package

Title Doing Dialectometry in R

Version 1.0

Date 2021-05-09

Description Quantitative approaches to dialectology based primarily on modifications of edit distance, which is more commonly known as the field of dialectometry. For further reference on the school of thought associated with these methods, see Wieling & Nerbonne (2015), <https://www.annualreviews.org/doi/10.1146/annurev-linguist-030514-124930>.

License GPL (>= 2)

Imports Rcpp (>= 1.0.5), ggplot2, deldir, ggvoronoi, dtw, reticulate, sf, dplyr, grDevices, tibble

LinkingTo Rcpp, RcppProgress

SystemRequirements C++11

LazyData true

Config/reticulate list(packages = list(list(package = ``numpy'', pip = TRUE), list(package = ``sklearn'', pip = TRUE), list(package = ``scipy'', pip = TRUE), list(package = ``python_speech_features'', pip = TRUE), list(package = ``speechpy'', pip = TRUE),))

RoxygenNote 7.1.1

Encoding UTF-8

Depends R (>= 2.10)

NeedsCompilation yes

Author Soh-Eun Shim [aut, cre]

Maintainer Soh-Eun Shim <soh-eun.shim@student.uni-tuebingen.de>

Repository CRAN

Date/Publication 2021-05-20 07:40:02 UTC

R topics documented:

`dialectR-package` *A short title line describing what the package does*

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
  ## Not run:
     ## Optional simple examples of the most important functions
     ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

`acoustic_distance` *Acoustic distance based on Mel-Frequency Cepstral Coefficients*

Description

This function implements an acoustic distance based on Mel-Frequency Cepstral Coefficients, upon which dynamic time warping is used to produce the results, as proposed by Bartelds et al. (2020). With an input of two audio files in the Waveform Audio File Format (i.e. wav), the function will return a distance between the two audios.

Usage

```
acoustic_distance(file1, file2)
```

Arguments

`file1`	The file to compare, which should be in the Waveform Audio File Format (i.e. wav).
`file2`	The other audio file to compare against, again as a wav.

Value

A number, indicating the distance between the two audio files.

References

Bartelds, M., Richter, C., Liberman, M., and Wieling, M. 2020. A New Acoustic-Based Pronunciation Distance Measure. *Frontiers in Artificial Intelligence*, 3, May. doi: 10.3389/frai.2020.00039

Examples

```
# Example 1: The acoustic distance between i and e

i_audio <- system.file("extdata", "i.wav", package="dialectR")
e_audio <- system.file("extdata", "e.wav", package="dialectR")
try(acoustic_distance(i_audio, e_audio),
 message("Python not available for testing"),
 silent = TRUE)
```

cluster_map *Visualize dialect groups with clustering methods*

Description

Input a distance matrix, upon which clustering will be performed and projected unto a map.

Usage

```
cluster_map(dist_mat, kml_points, kml_polygon, cluster_num, method)
```

Arguments

dist_mat	A distance matrix.
kml_points	A dataframe of kml (Keyhole Markup Language) points, as retrieved by get_points.
kml_polygon	A dataframe of kml polygons, as retrieved by get_polygons.
cluster_num	Number of clusters.
method	The agglomeration method that is passed to hclust. This can be chosen from the following: "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

Value

A map upon which dialect areas are clustered.

Examples

```
# Example 1: A cluster map of Dutch dialects
data(distDutch)
dutch_points <- get_points(system.file("extdata", "DutchKML.kml", package="dialectR"))
dutch_polygons <- get_polygons(system.file("extdata", "DutchKML.kml", package="dialectR"))
cluster_map(distDutch[1:100,1:100], dutch_points, dutch_polygons, 5, "ward.D2")
```

distance_matrix *Distance matrix for Dialectometry*

Description

Computes a distance matrix between dialect varieties, the results of which may be used for further analyses and plotting.

Usage

```
distance_matrix(
  dialect_data,
  funname,
  alignment_normalization = FALSE,
  delim = NULL
)
```

Arguments

dialect_data	A dataframe of dialect data, transcribed in the International Phonetic Alphabet.
funname	The distance metric to be used. This can be chosen from the following: "leven", "vc_leven".
alignment_normalization	A logical value, indicating whether or not the distance scores should be normalized by alignment length.
delim	An optional delimiter, in situations where multiple responses exist in the data.

Value

A distance matrix, where the values are the difference between dialects based on edit distance.

Examples

```
data(Dutch)
Dutch <- Dutch[1:3,1:3]
distance_matrix(Dutch, funname = "vc_leven", alignment_normalization = TRUE)
```

---

distDutch *Results of applying VC-levenshtein distance on the Dutch dataset*

Description

A dataset containing the results of applying VC-levenshtein distance on the Dutch dataset, which in turn is retrieved from the Goeman-Taeldeman-Van Reenen-Project.

Usage

```
distDutch
```

Format

A dataframe with 613 rows and 613 variables

Source

https://gabmap.nl/examples/

References

https://www.meertens.knaw.nl/projecten/mand/EGTRPdatata.html

`distmat_to_df` *Convert a dialectometric distance matrix to a dataframe*

Description

Input a distance matrix, with which a dataframe will be returned with the three columns of "row", "col", and "dist". The first two correspond with the rows and columns in the distance matrix, and the last refers to their crossing point, where the distance between them is given.

Usage

```
distmat_to_df(dist_matrix)
```

Arguments

`dist_matrix`	A distance matrix.

Value

A dataframe with the columns "row", "col", and "dist".

Examples

```
# Example 1: Dutch distance matrix to Dutch dataframe
data(distDutch)
distmat_to_df(distDutch)
```

`Dutch` *Dutch dialect data from the Goeman-Taeldeman-Van Reenen-Project*

Description

This dataset contains dialect data transcribed in the international phonetic alphabet, which is originally from the Goeman-Taeldeman-Van Reenen-Project.

Usage

```
Dutch
```

Format

A dataframe with 613 rows and 562 variables

Source

https://gabmap.nl/examples/

References

https://www.meertens.knaw.nl/projecten/mand/EGTRPdatata.html

DutchKML *KML (Keyhole Markup Language) file for the Dutch dataset*

Description

This is a KML file which marks the locations where the Dutch dataset was collected.

Format

KML file.

Source

https://gabmap.nl/examples/

See Also

https://www.meertens.knaw.nl/projecten/mand/EGTRPdatata.html

Examples

```
dutch_points <- get_points(system.file("extdata", "DutchKML.kml", package="dialectR"))
dutch_points
```

---

e *Wav file of "e" in the international phonetic alphabet*

Description

This is a wav file which contains the pronunciation of "e" in the international phonetic alphabet.

Format

wav file.

Source

http://www.phonetics.ucla.edu/course/chapter1/vowels.html

Examples

```
i_audio <- system.file("extdata", "i.wav", package="dialectR")
e_audio <- system.file("extdata", "e.wav", package="dialectR")
try(acoustic_distance(i_audio, e_audio),
 message("Python not available for testing"),
 silent = TRUE)
```

`get_clusters` *Clustering groups returned as dataframe*

Description

Input a distance matrix and returns a dataframe with two columns: area and clustering grouping, where a choice of clustering method is provided.

Usage

```
get_clusters(dist_mat, cluster_num, method)
```

Arguments

`dist_mat`	A distance matrix.
`cluster_num`	Number of clusters.
`method`	The agglomeration method that is passed to `hclust`. This can be chosen from the following: "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

Value

A map upon which dialect areas are clustered.

A dataframe with the two columns area and (clustering) grouping.

Examples

```
# Example 1:
data(distDutch)
get_clusters(distDutch, 5 ,"ward.D2")
```

get_points *Get KML points from KML data*

Description

Input a KML file path to get KML points data

Usage

```
get_points(kml_file_path)
```

Arguments

kml_file_path	A file path to a KML file.

Value

A dataframe with the columns (area) name, longitude, and latitude.

Examples

```
dutch_points <- get_points(system.file("extdata", "DutchKML.kml", package="dialectR"))
dutch_points
```

get_polygons *Get KML polygon from KML data*

Description

Input a KML file path to get KML polygon data

Usage

```
get_polygons(kml_file_path)
```

Arguments

kml_file_path	A file path to a KML file.

Value

A dataframe with the columns (area) name, longitude, and latitude.

Examples

```
dutch_polygons <- get_polygons(system.file("extdata", "DutchKML.kml", package="dialectR"))
dutch_polygons
```

i — *Wav file of "i" in the international phonetic alphabet*

Description

This is a wav file which contains the pronunciation of "i" in the international phonetic alphabet.

Format

wav file.

Source

http://www.phonetics.ucla.edu/course/chapter1/vowels.html

Examples

```
i_audio <- system.file("extdata", "i.wav", package="dialectR")
e_audio <- system.file("extdata", "e.wav", package="dialectR")
try(acoustic_distance(i_audio, e_audio),
 message("Python not available for testing"),
 silent = TRUE)
```

leven — *Edit distance for Dialectometry*

Description

An edit distance for use in Dialectometry. Allows for normalization by dividing alignment length, and for accommodating multiple responses with Bilbao distance, as proposed by Aurrekoetxea et al (2020).

Usage

```
leven(vec1, vec2, alignment_normalization = FALSE, delim = NULL)
```

Arguments

vec1	A vector of words.
vec2	A vector of words to be compared against.
alignment_normalization	A logical value, indicating whether or not the difference scores are to be normalized by alignment length.
delim	An optional delimiter, in situations where multiple responses exist in the data.

Value

A number indicating the number of operations to transform a string to the other, which optionally may undergo length normalization.

References

Aurrekoetxea, G., Nerbonne, J., and Rubio, J. 2020. Unifying Analyses of Multiple Responses. *Dialectologia*, 25:59–86.

Examples

```
leven("hit", "hot/hit", alignment_normalization = TRUE, delim = "/")
```

`mds_map` *Visualize dialect continua with MDS maps*

Description

Input a distance matrix and kml data, where multidimensional scaling will be applied on the former and projected onto a map.

Usage

```
mds_map(dist_mat, kml_points, kml_polygon)
```

Arguments

`dist_mat`	A distance matrix.
`kml_points`	A dataframe of kml points, as retrieved by `get_points`.
`kml_polygon`	A dataframe of kml polygons, as retrieved by `get_polygons`.

Value

A map upon which the results of multidimensional scaling are projected upon.

Examples

```
# Example 1: An MDS map of Dutch dialects
data(distDutch)
dutch_points <- get_points(system.file("extdata", "DutchKML.kml", package="dialectR"))
dutch_polygons <- get_polygons(system.file("extdata", "DutchKML.kml", package="dialectR"))
mds_map(distDutch, dutch_points, dutch_polygons)
```

vc_leven *VC-sensitive edit distance for Dialectometry*

Description

An edit distance that is sensitive to vowel and consonant alignment. If the aligned segments are a vowel-consonant pair, the difference is penalized as a score of 2; if not, 1. Allows for normalization by dividing alignment length, and for accommodating multiple responses with Bilbao distance, as proposed by Aurrekoetxea et al (2020).

Usage

```
vc_leven(vec1, vec2, alignment_normalization = FALSE, delim = NULL)
```

Arguments

vec1	A vector of words.
vec2	A vector of words to be compared against.
alignment_normalization	A logical value, indicating whether or not the difference scores are to be normalized by alignment length.
delim	An optional delimiter, in situations where multiple responses exist in the data.

Value

A number indicating the number of operations to transform a string to the other, which optionally may undergo length normalization.

References

Aurrekoetxea, G., Nerbonne, J., and Rubio, J. 2020. Unifying Analyses of Multiple Responses. *Dialectologia*, 25:59–86.

Examples

```
vc_leven("hit", "hot/hit", alignment_normalization = TRUE, delim = "/")
```

Index