Package 'multimolang'

December 2, 2024

Type Package

Title Multimodal Language Analysis Tools

Version 0.1.0

Description Tools for analyzing multimodal language data. Includes dfMaker for processing OpenPose outputs.

License AGPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports arrow,

Suggests roxygen2 (>= 7.0.0), testthat (>= 3.0.0), knitr, rmarkdown

URL https://github.com/daedalusLAB/multimolang.git

BugReports https://github.com/daedalusLAB/multimolang/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Keywords multimodal, language, OpenPose, analysis

NeedsCompilation no

Author Brian Herreño Jiménez [aut, cre]

Maintainer Brian Herreño Jiménez <brian.herreno@um.es>

R topics documented:

dfMaker *dfMaker Function*

Description

dfMaker processes and organizes keypoints data generated by OpenPose, compiling multiple JSON files into a structured data frame suitable for further analysis, visualization, or machine learning applications.

Usage

```
dfMaker(
  input.folder,
  config.path,
  output.file = NULL,
  output.path = NULL,
  no_save = FALSE,
  fast_scaling = TRUE,
  transformation_coords = c(1, 1, 5, 5)
)
```

Arguments

input.folder	Path to the folder containing OpenPose JSON files. The folder should contain only the JSON files to be processed; including other files may lead to unexpected behavior.
config.path	Path to the configuration file used for extracting metadata from filenames when processing data from the UCLA NewsScape archive. This configuration file specifies which metadata fields to extract based on the standard NewsScape filename format. If not provided, default settings are used, which may not extract NewsScape-specific metadata.
output.file	Name of the output file. If NULL and there is only one unique id in the data, an auto-generated name is used. If there are multiple unique id values and output.file is NULL, the function will generate an error; you must provide an explicit output file name.
output.path	Path to save the output file. If NULL, the file is saved in a default directory called df_outputs in the current working directory.
no_save	Logical. If TRUE, the output is not saved to a file.
fast_scaling	Logical. If TRUE, uses fast scaling for transformation. **Warning:** When fast_scaling is TRUE, scaling is performed only using pose keypoints (t_typepoint = 1), and the secondary vector v.j is not utilized.
transformation_coords	Numeric vector of length 4 that specifies the transformation coordinates. Each element of the vector has a specific role: 1. t_typepoint: Type of keypoints to use. Possible values:

   (a) 1: Body keypoints (pose).
   (b) 2: Face keypoints.
   (c) 3: Left hand keypoints.
   (d) 4: Right hand keypoints.
2. o_point: Index of the keypoint used as the **origin** in the new coordinate system.
3. i_point: Index of the keypoint that defines the primary base vector v.i.
4. j_point: Index of the keypoint that defines the secondary base vector v.j.
   - If i_point == j_point, v.j is calculated as a **perpendicular** vector to v.i (orthonormalization).
   - If fast_scaling = TRUE, v.j is not utilized and should be set to NA.

Details

This function depends on the **arrow** package for reading and writing JSON and Parquet files. Please ensure that the **arrow** package is installed.

When processing data with multiple unique id values, ensure that you provide an explicit output.file name to avoid errors.

The function expects JSON files generated by OpenPose with a specific structure. Variations in the OpenPose configuration or version may affect the format of these files. Ensure that the JSON files conform to the expected structure for accurate processing.

Each row in the data frame corresponds to a single keypoint detected in a specific frame. The columns **nx** and **ny** represent transformed coordinates based on the selected origin and scaling parameters. The **id** column associates the data with the input file from which the keypoints were extracted.

The data frame may contain missing values (NA) for keypoints that could not be reliably detected.

Value

A data frame containing the processed keypoints data with the following structure:

id Character. Identifier derived from the name of the file processed.

frame Numeric. Frame number from which the data is extracted.

people_id Integer. Identifier for each detected person in the frame.

type_points Character. Type of keypoints (e.g., "pose_keypoints", "face_keypoints", "hand_left_keypoints", "hand_right_keypoints").

points Integer. Index of the keypoints sequence.

x Numeric. X-coordinate of the keypoint.

y Numeric. Y-coordinate of the keypoint.

c Numeric. Confidence score for the detected keypoint, ranging from 0 to 1.

nx Numeric. Transformed X-coordinate in the new coordinate system.

ny Numeric. Transformed Y-coordinate in the new coordinate system.

Example of the returned data frame:

```
'data.frame': 8220 obs. of 10 variables:
 $ id      : chr  "2006-01-14_0600_US_KTTV-FOX_Ten_OClock_News_273-275_back_then" ...
 $ frame      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ people_id  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ type_points: chr  "pose_keypoints" "pose_keypoints" "pose_keypoints" "pose_keypoints" ...
 $ points     : int  0 1 2 3 4 5 6 7 8 9 ...
 $ x          : num  223 209 131 113 NA ...
 $ y          : num  113 178 178 273 NA ...
 $ c          : num  0.892 0.7 0.522 0.273 0 ...
 $ nx         : num  0.165 0 -0.945 -1.165 NA ...
 $ ny         : num  0.791 0 -0.000446 -1.14343 NA ...
```

R Version Requirements

This function uses the native pipe operator `|>` introduced in R version 4.1.0. Therefore, R version 4.1.0 or higher is required to use this function.

Error Handling

If the JSON files do not have the expected format or are empty, the function will skip these files and print a message indicating the issue. If `output.file` is `NULL` and multiple unique `id` values are found, the function will stop and generate an error, prompting you to provide an explicit `output.file` name.

Note on NewsScape Data

The **UCLA NewsScape** archive is the largest TV news video archive globally, containing over 300,000 news programs from the United States and around the world, dating from 1979 to the present. The collection provides streaming of news content and includes time-stamped closed-caption texts of most broadcasts, along with various metadata generated by machine learning and computer vision classifiers. This offers advanced search functions and enables new possibilities for teaching, research, and scholarship.

When processing data from NewsScape, the `config.path` parameter allows you to specify a configuration file that defines how to extract metadata from the filenames of the videos. The filenames in this archive have specific structures containing metadata such as date, time, country code, network code, program name, and time range.

Example of Configuration File (`config.json`)**:**

```
{
    "extract_datetime": true,
    "extract_time": true,
    "extract_exp_search": true,
    "extract_country_code": true,
    "extract_network_code": true,
    "extract_program_name": true,
    "extract_time_range": true,
    "timezone": "America/Los_Angeles"
}
```

This configuration allows you to control which metadata fields are extracted from the filenames. The search_exp variable is used in linguistic studies to analyze specific expressions within the content.

Important: Ensure that your data filenames follow the standard NewsScape naming convention for accurate metadata extraction. If your data does not conform to this naming convention, you may need to adjust your filenames or modify the configuration accordingly.

Example of a NewsScape Filename and Its Components:

```
2006-01-14_0600_US_KTTV-FOX_Ten_OClock_News_273-275_back_then_000000000000_keypoints.json
```

Breakdown of Filename Components:

2006-01-14 Extracted by extract_datetime: The date of the broadcast (YYYY-MM-DD).

0600 Extracted by extract_time: The time of the broadcast in 24-hour format (HHMM).

US Extracted by extract_country_code: The country code where the broadcast originated.

KTTV-FOX_Ten_OClock_News_273-275_back_then Extracted by extract_network_code, extract_program_name, and extract_exp_search:

- KTTV-FOX The network code and station identifier.
- Ten_OClock_News The program name.
- 273-275 The time range or segment identifier within the program.
- back_then Extracted by extract_exp_search: A specific expression or keyword used in linguistic studies.

Note on Timezone: The timezone parameter in the configuration file is used to standardize the broadcast times and is not treated as a variable within the data extraction process.

Example of how timezone is applied in R:

```
datetime <- as.POSIXct(datetime_str, format = "
```

References

For more information about the UCLA NewsScape archive, visit the official website: https://bigdatasocialscience.ucla.edu/newsscape/

The **arrow** R package is used for efficient reading and writing of JSON and Parquet files. For more information, visit: https://cran.r-project.org/package=arrow

For details on OpenPose, refer to: Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 172–186. https://doi.org/10.1109/TPAMI.2019.2929257

Alternatively, explore the OpenPose GitHub repository for implementation details: https://github.com/CMU-Perceptual-Computing-Lab/openpose

Examples

```
# Example 1: Define paths to example data included with the package
input.folder <- system.file("extdata/ex_videos/o1",
                            package = "multimolang")
output.file <- file.path(tempdir(), "processed_data.csv")
output.path <- tempdir()  # Use a temporary directory for writing output

# Run dfMaker with example data
df <- dfMaker(
  input.folder = input.folder,
  output.file = output.file,
  output.path = output.path,
  no_save = FALSE,
  fast_scaling = TRUE,
  transformation_coords = c(1, 1, 5, 5)
)

# View the first few rows of the resulting data frame
head(df)

# Example 2: Using NewsScape data with a custom configuration file

# Define paths to example data
input.folder <- system.file("extdata/ex_videos/o1",
                            package = "multimolang")

# Define the configuration file path
config.path <- system.file("extdata/config_all_true.json",
                           package = "multimolang")

# Define output paths
output.file <- file.path(tempdir(), "processed_data.csv")
output.path <- tempdir()

# Run dfMaker with custom configuration
df <- dfMaker(
  input.folder = input.folder,
  config.path = config.path,
  output.file = output.file,
  output.path = output.path,
  no_save = FALSE,
  fast_scaling = TRUE,
  transformation_coords = c(1, 1, 5, 5)
)

# View the first few rows
head(df)
```

Index