Package 'ConversationAlign'

July 22, 2025

Type Package

Title Process Text and Compute Linguistic Alignment in Conversation Transcripts

Version 0.2.0

Maintainer Jamie Reilly <jamie_reilly@temple.edu>

Description Imports conversation transcripts into R, concatenates them into a single dataframe appending event identifiers, cleans and formats the text, then yokes user-specified psycholinguistic database values to each word. 'ConversationAlign' then computes alignment indices between two interlocutors across each transcript for >40 possible semantic, lexical, and affective dimensions. In addition to alignment, 'ConversationAlign' also produces a table of analytics (e.g., token count, type-token-ratio) in a summary table describing your particular text corpus.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5)

Imports DescTools, dplyr (>= 0.4.3), httr, magrittr, purrr, rlang, stringi, stringr, textstem, tibble, tidyr, tidyselect, stats, utils, YRmisc, zoo

Suggests devtools, knitr, rmarkdown, testthat (>= 3.0.0)

URL https://github.com/Reilly-ConceptsCognitionLab/ConversationAlign

RoxygenNote 7.3.2

LazyData true

VignetteBuilder knitr

Collate 'ConversationAlign-package.R' 'compute_auc.R' 'compute_lagcorr.R' 'corpus_analytics.R' 'data.R' 'globals.R' 'prep_dyads.R' 'read_1file.R' 'read_dyads.R' 'replacements_25.R' 'summarize_dyads.R' 'utils.R' 'zzz.R'

Config/testthat/edition 3

NeedsCompilation no

Author Jamie Reilly [aut, cre] (ORCID: <https://orcid.org/0000-0002-0891-438X>),

Virginia Ulichney [aut],
Ben Sacks [aut],
Sarah Weinstein [ctb],
Chelsea Helion [ctb],
Gus Cooney [ctb]

Repository CRAN

Date/Publication 2025-07-22 10:10:16 UTC

Contents

`corpus_analytics` *corpus_analytics*

Description

Produces a table of corpus analytics including numbers of complete observations at each step, word counts, lexical diversity (e.g., TTR), stopword ratios, etc. Granularity of the summary statistics are guided by the user (e.g., by conversation, by conversation and speaker, collapsed all)

Usage

```
corpus_analytics(dat_prep)
```

Arguments

`dat_prep`	takes dataframe produced from the df_prep() function

Value

dataframe with summary statistics (mean, SD, range) for numerous corpus analytics (e.g., token count, type-token-ratio, word-count-per-turn) for the target conversation corpus. Summary data structured in table format for easy export to a journal method section.

load_github_data *Load all .rda files from a GitHub data folder into the package environment*

Description

Load all .rda files from a GitHub data folder into the package environment

Usage

```
load_github_data(
  repo = "Reilly-ConceptsCognitionLab/ConversationAlign_Data",
  branch = "main",
  data_folder = "data",
  envir = parent.frame()
)
```

Arguments

repo	GitHub repository (e.g., "username/repo")
branch	Branch name (default: "main")
data_folder	Remote folder containing .rda files (default: "data/")
envir	Environment to load into (default: package namespace)

Value

nothing, loads data (as rda files) from github repository needed for other package functions

MaronGross_2013 *Sample Dyadic Interview Transcript: Marc Maron and Terry Gross Radio Interview 2013*

Description

Text and talker information delineated, raw transcript, multiple lines per talker

Usage

```
MaronGross_2013
```

Format

"MaronGross_2013" A data.frame with 546 obs, 2 vars:

text text from interview

speaker speaker identity ...

`NurseryRhymes` *Text and talker information delineated, 3 separate nursery rhymes, good for computing analytics and word counts*

Description

Text and talker information delineated, 3 separate nursery rhymes, good for computing analytics and word counts

Usage

```
NurseryRhymes
```

Format

"NurseryRhymes" A data.frame with 100 observations, 2 vars:

Event_ID factor 3 different simulated conversations

Participant_ID fictional speaker names, 2 each conversation

Text_Raw simulated language production, actually looped phrases from nursery rhymes ...

`NurseryRhymes_Prepped` *Text and talker information delineated, 3 separate nursery rhymes, good for computing analytics and word counts*

Description

Text and talker information delineated, 3 separate nursery rhymes, good for computing analytics and word counts

Usage

```
NurseryRhymes_Prepped
```

Format

"NurseryRhymes_Prepped" A data.frame with 1507 x 7 observations, 5 vars:

Event_ID factor 3 different simulated conversations

Participant_ID fictional speaker names, 2 each conversation

Exchange_Count sequential numbering of exchanges by conversation, 1 exchange = 2 turns

Turn_Count sequential numbering of turns by conversation

Text_Clean content words

emo_anger raw value of anger salience yoked to each word ...

prep_dyads *prep_dyads*

Description

Cleans, vectorizes and appends lexical norms to all content words in a language corpus. User guides options for stopword removal and lemmatization. User selects up to three psycholinguistic dimensions to yoke norms on each content word in the original conversation transcript.

Usage

```
prep_dyads(
  dat_read,
  lemmatize = TRUE,
  omit_stops = TRUE,
  which_stoplist = "Temple_stops25",
  verbose = TRUE
)
```

Arguments

dat_read	dataframe produced from read_dyads() function
lemmatize	logical, should words be lemmatized (switched to base morphological form), default is TRUE
omit_stops	option to remove stopwords, default TRUE
which_stoplist	user-specified stopword removal method with options including "none", "SMART", "MIT_stops", "CA_OriginalStops", or "Temple_Stopwords25". "Temple_Stopwords25 is the default list
verbose	display detailed output such as error messages and progress (default is TRUE)

Value

dataframe with text cleaned and vectorized to a one word per-row format. Lexical norms and metadata are appended to each content word. Cleaned text appears under a new column called 'Text_Clean'. Any selected dimensions (e.g., word length) and metadata are also appended to each word along with speaker identity, turn, and Event_ID (conversation identifier).

`read_1file` *read_1file*

Description

Reads pre-formatted dyadic (2 interlocutor) conversation transcript already imported into your R environment.

Usage

```
read_1file(my_dat)
```

Arguments

`my_dat`	one conversation transcript already in the R environment

Value

a dataframe formatted with 'Event_ID', "Participant_ID", "Text_Raw" fields – ready for clean_dyads()

`read_dyads` *read_dyads*

Description

Reads pre-formatted dyadic (2 interlocutor) conversation transcripts from your machine. Transcripts must be either csv or txt format. IF you are supplying a txt file, your transcript must be formatted as an otter.ai txt file export. Your options for using csv files are more flexible. ConversationAlign minimally requires a csv file with two columns, denoting interlocutor and text. Each separate conversation transcript should be saved as a separate file. ConversationAlign will use the file names as a document ID. Within the read dyads function, set the my_path argument as the directory path to the local folder containing your transcripts on your machine (e.g., "my_transcripts"). Please see our github page for examples of properly formatted transcripts: https://github.com/Reilly-ConceptsCognitionLab/ConversationAl

Usage

```
read_dyads(my_path = "my_transcripts")
```

Arguments

`my_path`	folder of conversation transcripts in csv or txt format

Value

a dataframe where each individual conversation transcript in a user's directory has been concatenated. read_dyads appends a unique document identifier to each conversation transcript appending its unique filename as a factor level to 'Event_ID'.

summarize_dyads *summarize_dyads*

Description

Calculates and appends 3 measures for quantifying alignment. Appends the averaged value for each selected dimension by turn and speaker. Calculates and Spearman's rank correlation between interlocutor time series and appends by transcript. Calculates the area under the curve of the absolute difference time series between interlocutor time series. The length of the difference time series can be standardized the shortest number of exchanges present in the group using an internally defined resampling function, called with resample = TRUE. Spearman's rank correlation and area under the curve become less reliable for dyads under 30 exchanges.

Usage

```
summarize_dyads(
  df_prep,
  custom_lags = NULL,
  sumdat_only = TRUE,
  corr_type = "Pearson"
)
```

Arguments

df_prep	produced in the align_dyads function
custom_lags	integer vector, should any lags be added in addition to -2, 0, 2
sumdat_only	default=TRUE, group and summarize data, two rows per conversation, one row for each participant, false will fill down summary statistics across all exchanges
corr_type	option for computing lagged correlations turn-by-turn covariance (default='Pearson')

Value

either: - a grouped dataframe with summary data aggregated by converation (Event_ID) and participant if sumdat_only=T. - the origoinal dataframe 'filled down' with summary data (e.g., AUC, turn-by-turn correlations) for each conversation is sumdat_only=F.

Index