Package: SemanticDistance (via r-universe)

October 12, 2026

Type Package

Version 0.1.1

Title Compute Semantic Distance Between Text Constituents

Maintainer Jamie Reilly <jamie_reilly@temple.edu>

Description Cleans and formats language transcripts guided by a series of transformation options (e.g., lemmatize words, omit stopwords, split strings across rows). 'SemanticDistance' computes two distinct metrics of cosine semantic distance (experiential and embedding). These values reflect pairwise cosine distance between different elements or chunks of a language sample. 'SemanticDistance' can process monologues (e.g., stories, ordered text), dialogues (e.g., conversation transcripts), word pairs arrayed in columns, and unordered word lists. Users specify options for how they wish to chunk distance calculations. These options include: rolling ngram-to-word distance (window of n-words to each new word), ngram-to-ngram distance (2-word chunk to the next 2-word chunk), pairwise distance between words arrayed in columns, matrix comparisons (i.e., all possible pairwise distances between words in an unordered list), turn-by-turn distance (talker to talker in a dialogue transcript). 'SemanticDistance' includes visualization options for analyzing distances as time series data and simple semantic network dynamics (e.g., clustering, undirected graph network).

License LGPL (>= 3)

Encoding UTF-8

LazyData true

LazyDataCompression xz

URL https://github.com/Reilly-ConceptsCognitionLab/SemanticDistance, https://reilly-conceptscognitionlab.github.io/SemanticDistance/

BugReports https://github.com/Reilly-ConceptsCognitionLab/SemanticDistance/issues

Depends R (>= 3.5)

Imports ape, cluster, dendextend, dplyr, graphics, httr, igraph, lsa, magrittr, purrr, rlang, stats, stringi, stringr, textstem, tidyselect, tm, tidyr, textclean, tools, utils, wesanderson,

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

Collate 'clean_dialogue.R' 'clean_monologue_or_list.R' 'clean_paired_cols.R' 'data.R' 'dist_anchor.R' 'dist_dialogue.R' 'dist_ngram2ngram.R' 'dist_ngram2word.R' 'dist_paired_cols.R' 'globals.R' 'reexports.R' 'replacements_25.R' 'utils.R' 'wordlist_to_network.R' 'zzz.R'

Config/testthat/edition 3

Repository https://reilly-conceptscognitionlab.r-universe.dev

Date/Publication 2026-10-12 13:54:01 UTC

RemoteUrl https://github.com/reilly-conceptscognitionlab/semanticdistance

RemoteRef HEAD

RemoteSha 25e7a9d3ab13cf8fd0eeddafe8eb20144b301585

Contents

clean_dialogue . . . 3
clean_monologue_or_list . . . 3
clean_paired_cols . . . 4
Dialogue_Typical . . . 5
dist_anchor . . . 5
dist_dialogue . . . 6
dist_ngram2ngram . . . 6
dist_ngram2word . . . 7
dist_paired_cols . . . 7
glowca_25 . . . 8
Grandfather_Passage . . . 8
load_github_data . . . 9
Monologue_Typical . . . 9
SD15_2025_complete . . . 10
Temple_stops25 . . . 11
Unordered_List . . . 11
Word_Pairs . . . 12
wordlist_to_network . . . 12

Index **14**

clean_dialogue *clean_dialogue*

Description

Cleans a transcript where there are two or more talkers. User specifies the dataframe and column name where target text is stored in addition a factor variable corresponding to the identity of the person producing corresponding text. Users also specify cleaning parameters for stopword removal and lemmatization (both defaulting to TRUE). Function splits and unlists text so that the output is in a one-row-per-word format marked by a unique numeric identifier (i.e., 'id_orig'). Function appends a turn_count sequence used for aggregating all the words within each turn. If a speaker generates no complete observations because of stopword removal, the turn counter will not increment until a talker switch AND a complete observation is observed.

Usage

```
clean_dialogue(dat, wordcol, who_talking, omit_stops = TRUE, lemmatize = TRUE)
```

Arguments

dat	a datataframe with at least one target column of string data
wordcol	quoted column name storing the strings that will be cleaned and split
who_talking	quoted column name with speaker/talker identities will be factorized
omit_stops	T/F user wishes to remove stopwords (default is TRUE)
lemmatize	T/F user wishes to lemmatize each string (default is TRUE)

Value

a dataframe

---

clean_monologue_or_list
clean_monologue_or_list

Description

Cleans and formats text. User specifies the dataframe and column name where target text is stored as arguments to the function. Default option is to lemmatize strings. Function splits and unlists text so that the output is in a one-row-per-word format marked by a unique numeric identifier (i.e., 'id_orig')

Usage

```
clean_monologue_or_list(dat, wordcol, omit_stops = TRUE, lemmatize = TRUE)
```

Arguments

dat	a dataframe with at least one target column of string data
wordcol	quoted column name storing the strings that will be cleaned and split
omit_stops	option for omitting stopwords default is TRUE
lemmatize	option for lemmatizing strings default is TRUE

Value

a dataframe

---

clean_paired_cols *clean_paired_cols*

Description

Cleans a transcript where word pairs are arrayed in two columns.

Usage

```
clean_paired_cols(dat, wordcol1, wordcol2, lemmatize = TRUE)
```

Arguments

dat	a dataframe with two columns of words you want pairwise distance for
wordcol1	quoted column name storing the first string for comparison
wordcol2	quoted column name storing the second string for comparison
lemmatize	T/F user wishes to lemmatize each string (default is TRUE)

Value

a dataframe

Dialogue_Typical *A Typical Dialogue Transcript*

Description

A sample dyadic conversation transcript where two people are conversing.

Usage

```
Dialogue_Typical
```

Format

"Dialogue_Typical" A data frame with 5 rows and 2 columns:

word fictional text from a language transcript

speaker Mary or Peter: fictional speaker identities ...

dist_anchor *dist_anchor*

Description

Function takes dataframe cleaned using 'clean_monologue', computes rolling chunk-to-chunk distance between user-specified ngram size (e.g., 2-word chunks)

Usage

```
dist_anchor(dat, anchor_size = 10)
```

Arguments

dat	a dataframe prepped using 'clean_monologue' fn
anchor_size	an integer specifying the number of words in the initial chunk for comparison to new words as the sample unfolds

Value

a dataframe

dist_dialogue *dist_dialogue*

Description

Function takes dataframe cleaned using 'clean_dialogue' and computes two metrics of semantic distance turn-to-turn indexing a 'talker' column. Sums all the respective semantic vectors within each tuern, cosine distance to the next turn's composite vector

Usage

```
dist_dialogue(dat, who_talking)
```

Arguments

dat	a dataframe prepped using 'clean_dialogue' fn with talker data and turncount appended
who_talking	factor variable with two levels specifying an ID for the person producing the text in 'word_clean'

Value

a dataframe

dist_ngram2ngram *dist_ngram2ngram*

Description

Function takes dataframe cleaned using 'clean_monologue', computes rolling chunk-to-chunk distance between user-specified ngram size (e.g., 2-word chunks)

Usage

```
dist_ngram2ngram(dat, ngram)
```

Arguments

dat	a dataframe prepped using 'clean_monologue' fn
ngram	an integer specifying the window size of words for computing distance to a target word

Value

a dataframe

dist_ngram2word *dist_ngram2word*

Description

Function takes dataframe cleaned using 'clean_monologue', computes two metrics of semantic distance for each word relative to the average of the semantic vectors within an n-word window appearing before each word. User specifies the window (ngram) size. The window 'rolls' across the language sample providing distance metrics

Usage

```
dist_ngram2word(dat, ngram)
```

Arguments

dat	a dataframe prepped using 'clean_monologue' fn
ngram	an integer specifying the window size of words for computing distance to a target word will go back skipping NAs until content words equals the ngram window

Value

a dataframe

dist_paired_cols *dist_paired_cols*

Description

Function takes dataframe cleaned using 'clean_2columns', computes two metrics of semantic distance for each word pair arrayed in Col1 vs. Col2

Usage

```
dist_paired_cols(dat)
```

Arguments

dat	a dataframe prepped using clean_2columns' with word pairs arrayed in two columns

Value

a dataframe

glowca_25 *Glove Semantic Embeddings*

Description

Word embeddings (300 hyperparameter dimensions, 59061 words). Each word is one row.

Usage

```
glowca_25
```

Format

"glowca_25" A data frame with 59061 observations of 301 variables

word word characterized across embeddings

Param_1 hyperparameter number 1

Param_300 hyperparameter number 300 ...

Grandfather_Passage *The Grandfather Passage: A Standardized Reading Passage*

Description

A monologue discourse sample. Grandfather Passage is a well-known test of reading aloud.

Usage

```
Grandfather_Passage
```

Format

"Grandfather_Passage" A data frame with 1 observation of 1 variable:

mytext text from the Grandfather Passage unsplit ...

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/SemanticDistance_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

Monologue_Typical *A Typical Monologue Transcript*

Description

Dataframe with ordered text squashed into a single cell.

Usage

```
Monologue_Typical
```

Format

"Monologue_Typical" A data frame with 1 row and 1 column

mytext text from a language transcript ...

SD15_2025_complete *SD15_2025_complete Experiential Semantic Distance Values*

Description

Word embeddings (300 dimensions, 59061 words). Each word is one row.

Usage

```
SD15_2025_complete
```

Format

"SD15_2025_complete" A data frame with 25,050 observations of 16 variables

word word characterized across 15 ratings

Param_auditory_z z-score of auditory salience from Lancaster Sensorimotor Norms

Param_gustatory_z z-score of gustatory salience from Lancaster Sensorimotor Norms

Param_haptic_z z-score of haptic salience from Lancaster Sensorimotor Norms

Param_interoceptive_z z-score of interoceptive salience from Lancaster Sensorimotor Norms

Param_visual_z z-score of visual salience from Lancaster Sensorimotor Norms

Param_olfactory_z z-score of olfactory salience from Lancaster Sensorimotor Norms

Param_handarm_z z-score of handarm motor salience from Lancaster Sensorimotor Norms

Param_excitement_z z-score of excitement salience from affectvec

Param_surprised_z z-score of surprise salience from affectvec

Param_fear_z z-score of fear salience from affectvec

Param_anger_z z-score of anger salience from affectvec

Param_disgust_z z-score of disgust salience from affectvec

Param_sadness_z z-score of sadness salience from affectvec

Param_happiness_z z-score of happiness salience from affectvec

Param_contempr_z z-score of contempt salience from affectvec ...

Temple_stops25 *Stopword List*

Description

List of stopwords

Usage

```
Temple_stops25
```

Format

"Temple_stops25" A data frame with 829 observations of 4 variables

id_orig numeric identifier

word stopword target

length length in words

pos universal part-of-speech tag ...

Unordered_List *Unordered_List*

Description

No talker delineated. List of 17 words spanning 4 semantic categories, Good for examining clustering

Usage

```
Unordered_List
```

Format

"Unordered_List" A data frame with 1 rows and 1 columns:

mytext unsplit list of words containing musical instruments, weapongs, fruits, emotions

Word_Pairs *Word Pairs in Columns*

Description

first target word for computing distance in one column, second word in another column.

Usage

```
Word_Pairs
```

Format

"Word_Pairs" A data frame with 27 rows and 2 columns:

word1 text corresponding to the first word in a pair to contrast

word2 text corresponding to the second word in a pair to contrast ...

wordlist_to_network *wordlist_to_network*

Description

Takes a vector of words with semantic distance ratings, converts to a square matrix, then to a euclidean distance matrix (all word pairs), then plots the words in either a cluster dendrogram or simple igraph network

Usage

```
wordlist_to_network(
  dat,
  wordcol,
  output = "dendrogram",
  dist_type = "embedding"
)
```

Arguments

dat	dataframe with text in it (cleaned using clean_monologue_or_list function
wordcol	quoted argument identifying column in dataframe with target text
output	quoted argument for type of output default is 'dendrogram', alternate is 'network'
dist_type	quoted argument semantic norms for running distance matrix on default='embedding', other is 'SD15'

Details

This function internally calls eval_kmeans_clustersize for cluster evaluation. The dendrogram visualization is based on hierarchical clustering of semantic distances.

Value

a plot of a dendrogram or an igraph network AND a cosine distance matrix

Index

∗ **datasets**
Dialogue_Typical, 5
glowca_25, 8
Grandfather_Passage, 8
Monologue_Typical, 9
SD15_2025_complete, 10
Temple_stops25, 11
Unordered_List, 11
Word_Pairs, 12

clean_dialogue, 3
clean_monologue_or_list, 3
clean_paired_cols, 4

Dialogue_Typical, 5
dist_anchor, 5
dist_dialogue, 6
dist_ngram2ngram, 6
dist_ngram2word, 7
dist_paired_cols, 7

glowca_25, 8
Grandfather_Passage, 8

load_github_data, 9

Monologue_Typical, 9

SD15_2025_complete, 10

Temple_stops25, 11

Unordered_List, 11

Word_Pairs, 12
wordlist_to_network, 12