Package: keyclust (via r-universe)

October 17, 2026

Type Package

Title A Model for Semi-Supervised Keyword Extraction from Word Embedding Models

Version 1.2.5

Description A fast and computationally efficient algorithm designed to enable researchers to efficiently and quickly extract semantically-related keywords using a fitted embedding model. For more details about the methods applied, see Chester (2025). <doi:10.17605/OSF.IO/5B7RQ>.

Encoding UTF-8

License GPL-3

Depends R (>= 4.1.0)

Imports data.table (>= 1.14.8), textstem (>= 0.1.4)

Suggests knitr, R.utils, rmarkdown, spelling, testthat

LinkingTo Rcpp

LazyData TRUE

LazyDataCompression xz

RoxygenNote 7.3.2

Language en-US

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libicu-dev libxml2-dev

Repository https://pchest.r-universe.dev

Date/Publication 2025-05-29 21:58:19 UTC

RemoteUrl https://github.com/pchest/keyclust

RemoteRef HEAD

RemoteSha d7cbec4502789be66707b263bf66532076e3a558

Contents

cosimilarity_matrix 2
keyclust 2
print.keyclust 3
process_embed 4
similarity_matrix 5
terms.keyclust 5
wordemb_FasttextEng_sample 6

Index **7**

`cosimilarity_matrix` *Returns cosine cosimilarity matrix for the terms generated by keyclust*

Description

A function that extracts the cosimilarity matrix for terms generated by `keyclust()`

Usage

```
cosimilarity_matrix(x)
```

Arguments

x	output from `keyclust()`

Value

An N x N matrix of cosine cosimilarity values, where n is the number of terms in the provided embedding model

---

`keyclust` *Algorithm designed to efficiently extract keywords from a cosine similarity matrix*

Description

This function takes a cosine similarity matrix derived from a word embedding model, along with a set of seed words and outputs a semantically-related set of keywords of a length and cosimilarity determined by the user

Usage

```
keyclust(
  sim_mat,
  seed_words,
  sim_thresh = 0.25,
  max_n = 50,
  dictionary = NULL,
  exclude = NULL,
  verbose = TRUE
)
```

Arguments

sim_mat	A cosine similarity matrix produced by cosine.
seed_words	A set of user-provided seed words that best represent the target concept.
sim_thresh	Minimum cosine similarity a candidate word must have to the existing set of keywords for it to be added.
max_n	The maximum size of the output set of keywords.
dictionary	An optional dictionary that maps metadata, such as definitions, to keywords.
exclude	A vector of words that the user does not want included in the final keyword set.
verbose	If true, keyclust will produce live updates as it adds keywords.

Value

A list containing a data frame of keywords and their cosine similarities, and a matrix of cosine similarities.

Examples

```
# Create a set of keywords using a pre-defined set of seeds
seeds <- c("october", "november")
# Create a cosine similarity matrix from a word embedding model
simmat_FasttextEng_sample <- wordemb_FasttextEng_sample |>
    process_embed(words='words') |>
    similarity_matrix(words = "words")
# Use keyclust to generate a set of keywords
months <- keyclust(simmat_FasttextEng_sample, seed_words = seeds, max_n = 8)
```

---

print.keyclust *Prints terms generated by keyclust*

Description

Prints terms generated by keyclust

Usage

```
## S3 method for class 'keyclust'
print(x, ...)
```

Arguments

x	output from keyclust()
...	additional arguments not used

Value

A message indicating the number of keywords produced and a preview of the first few keywords.

---

process_embed *A tool designed to reduce redundant terms in a fitted embedding model*

Description

Takes a fitted embedding model as an input. Allows users to combine embeddings by the case, stem, or lemma of associated terms.

Usage

```
process_embed(
  x,
  words = NULL,
  punct = TRUE,
  tolower = TRUE,
  lemmatize = TRUE,
  stem = FALSE
)
```

Arguments

x	A fitted word embedding model in the data frame format
words	The name of a column that corresponds to the word dimension of the fitted word embeddings
punct	Removes punctuation
tolower	Combines terms that differ by case
lemmatize	Combines terms that share a common lemma. Uses the lexicon package by default.
stem	Combines terms that share a common stem. *Note:* Stemming should not be used in conjunction with lemmatize.

Value

A data frame with the same columns as the input, but with redundant terms combined.

similarity_matrix *Algorithm designed to create a cosine similarity matrix from a fitted word embedding model*

Description

This function takes a fitted word embedding model and computes the cosine similarity between each word.

Usage

```
similarity_matrix(x, words = NULL, max_terms = 25000)
```

Arguments

x	A word embedding matrix
words	A vector of words or the name of a column that corresponds to the word dimension of the fitted word embeddings
max_terms	The maximum number of embedding terms that will be included in output similarity matrix. Assumes that embedding input is ordered by word frequency.

Value

An N x N matrix of cosine similarity scores between words from a fitted word embedding model.

Examples

```
# Create a set of keywords using a pre-defined set of seeds
simmat <- similarity_matrix(wordemb_FasttextEng_sample, words = "words")
```

terms.keyclust *Returns terms generated by keyclust*

Description

A function that returns the terms and their cosine cosimilarities produced by keyclust()

Usage

```
## S3 method for class 'keyclust'
terms(x, ...)
```

Arguments

x	output from keyclust()
...	additional arguments not used

Value

A data frame of terms and their cosine similarities.

`wordemb_FasttextEng_sample`

Sample from the pre-trained English fastText model

Description

This is a data frame containing the 2,000 most frequently occurring terms from Facebook's English-language fastText word embeddings model.

Usage

```
wordemb_FasttextEng_sample
```

Format

A 2000 row and 301 column data frame. The row represents the word embedding term, while the numeric columns represent the word embedding dimension. The character column gives the terms associated with each word vector.

References

P. Bojanowski*, E. Grave*, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information (arxiv)

Examples

```
data(wordemb_FasttextEng_sample)
head(wordemb_FasttextEng_sample)
```

Index

∗ **data**
    wordemb_FasttextEng_sample, 6
∗ **keyclust**
    cosimilarity_matrix, 2
    keyclust, 2
    print.keyclust, 3
    process_embed, 4
    similarity_matrix, 5
    terms.keyclust, 5

cosimilarity_matrix, 2

keyclust, 2
keyclust(), *2*, *4*, *5*

print.keyclust, 3
process_embed, 4

similarity_matrix, 5

terms.keyclust, 5

wordemb_FasttextEng_sample, 6