Package 'wordorientation'

August 8, 2026

Type Package

Title Detect Attraction and Repulsion Between Words in Text

Version 0.1.0

Description Provides tools to quantify how strongly pairs of words attract or repel each other in a text corpus, based on co-occurrence patterns. For each word pair, the phi coefficient (a correlation measure for binary variables) is computed from a document-term matrix and tested for significance, then classified as showing attraction (co-occurring more than chance would predict), repulsion (co-occurring less than chance would predict), or no significant relationship. A full pipeline is provided from raw text to a labeled network visualization. Unlike general-purpose pairwise correlation tools, 'wordorientation' is built specifically for text: it handles tokenization and stopword removal, applies significance-based classification rather than reporting a raw correlation coefficient alone, and produces a ready-to-plot attraction/repulsion network.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats, graphics, igraph, scales

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

RoxygenNote 7.3.1

URL https://github.com/yosleycarrero2025/wordorientation

BugReports https://github.com/yosleycarrero2025/wordorientation/issues

NeedsCompilation no

Author Yosley Carrero [aut, cre]

Maintainer Yosley Carrero <c3046338@newcastle.ac.uk>

Repository CRAN

Date/Publication 2026-08-08 14:40:02 UTC

Contents

`analyze_word_orientation` *Run the full word attraction/repulsion pipeline*

Description

Convenience wrapper that tokenizes text, computes co-occurrence counts, classifies word pairs as attraction/repulsion/neutral, and optionally plots the resulting network, in a single call.

Usage

```
analyze_word_orientation(
  data,
  text_col,
  doc_col = NULL,
  min_count = 5,
  alpha = 0.05,
  top_n = 15,
  stopwords = TRUE,
  custom_stopwords = NULL,
  plot = TRUE
)
```

Arguments

`data`	A data frame containing a text column.
`text_col`	Name of the column in `data` containing the text.
`doc_col`	Optional document id column; see `tokenize_posts`.
`min_count`	Minimum document frequency for a word to be included; see `cooccurrence_counts`.
`alpha`	Significance threshold; see `word_orientation`.
`top_n`	Number of top pairs per relationship type to plot; see `plot_orientation_network`.
`stopwords`	Logical; remove built-in English stopwords. Default `TRUE`.
`custom_stopwords`	Optional extra stopwords; see `tokenize_posts`.
`plot`	Logical; if `TRUE` (default), also build the network plot and include it in the result.

Value

A list with elements `tokens`, `cooccurrence`, `scored` (the full attraction/repulsion table), and, if `plot = TRUE`, `network_plot`.

Examples

```
result <- analyze_word_orientation(
  example_social_posts(),
  text_col = "text", doc_col = "id",
  min_count = 2, top_n = 10
)
head(result$scored)
```

`cooccurrence_counts` *Compute pairwise co-occurrence contingency counts*

Description

Builds a document-term incidence matrix from a token table and returns, for every pair of words that meet a minimum frequency threshold, the four cell counts of the 2x2 contingency table needed for a phi coefficient: `a` (both present), `b` (word1 only), `c` (word2 only), and `d` (neither present).

Usage

```
cooccurrence_counts(tokens, min_count = 5, max_vocab = 500)
```

Arguments

`tokens`	A token table as returned by `tokenize_posts`, with columns `doc_id` and `word`.
`min_count`	Minimum number of documents a word must appear in on its own to be included in the pairwise analysis. Rare words produce unstable phi estimates and are dropped by default.
`max_vocab`	Maximum number of most-frequent words to keep before forming pairs, to keep memory bounded for large corpora. Default 500.

Value

A data frame with columns `word1`, `word2`, `a`, `b`, `c`, `d`, and `n` (total number of documents).

Examples

```
df <- data.frame(id = 1:2, text = c("we are honest and fair",
                                    "they are corrupt and selfish"))
tokens <- tokenize_posts(df, text_col = "text", doc_col = "id")
cooccurrence_counts(tokens, min_count = 1)
```

example_social_posts *Example social media posts*

Description

A small built-in toy corpus of short posts, useful for demonstrating and testing the package's functions without needing an external data file. The posts are constructed so that some words show strong attraction (e.g. "we" and "honest") and others show strong repulsion (e.g. "we" and "they"), for illustration purposes only.

Usage

```
example_social_posts()
```

Value

A data frame with columns `id` (integer document id) and `text` (character).

Examples

```
example_social_posts()
```

plot_orientation_network *Plot a word attraction/repulsion network*

Description

Draws the strongest attraction and repulsion word pairs as a network graph, with edge color showing the direction of the relationship and edge width showing its strength. Built on base **igraph** plotting to keep the package's dependency footprint small.

Usage

```
plot_orientation_network(
  scored,
  top_n = 15,
  show = c("both", "repulsion", "attraction"),
  layout = NULL,
  ...
)
```

Arguments

scored	A data frame as returned by word_orientation, with columns word1, word2, phi, relationship.
top_n	Number of strongest edges (by absolute phi) to keep per relationship type. Default 15.
show	Which edges to draw: "both" (default), "repulsion", or "attraction".
layout	Layout function from **igraph** to use, e.g. igraph::layout_with_fr. Default NULL uses Fruchterman-Reingold.
...	Additional arguments passed on to plot.igraph.

Value

Invisibly, the igraph graph object that was plotted. Called for its plotting side effect.

---

tokenize_posts *Tokenize a text column into a long-format token table*

Description

Splits each document into lowercase word tokens, strips punctuation, and optionally removes stopwords. Returns a long-format data frame with one row per token, keeping a document identifier so co-occurrence can later be computed within documents.

Usage

```
tokenize_posts(
  data,
  text_col,
  doc_col = NULL,
  stopwords = TRUE,
  custom_stopwords = NULL
)
```

Arguments

data	A data frame containing at least a text column.
text_col	Name of the column in data containing the text.
doc_col	Optional name of a column to use as the document id. If NULL, row numbers of data are used as document ids.
stopwords	Logical; if TRUE (default), a built-in English stopword list is removed from the tokens.
custom_stopwords	Optional character vector of additional stopwords to remove, on top of (or instead of) the built-in list.

Value

A data frame with columns `doc_id` and `word`, one row per token.

Examples

```
df <- data.frame(id = 1:2, text = c("We are honest and fair",
                                    "They are corrupt and selfish"))
tokenize_posts(df, text_col = "text", doc_col = "id")
```

`word_orientation` *Classify word pairs as attracted, repelled, or unrelated*

Description

Computes the phi coefficient for every word pair in a contingency-count table, tests it for significance, and labels each pair as `"attraction"` (co-occurring significantly more than chance), `"repulsion"` (co-occurring significantly less than chance), or `"neutral"` (no significant relationship at the chosen threshold).

Usage

```
word_orientation(cooc, alpha = 0.05, p_adjust_method = "BH")
```

Arguments

`cooc`	A data frame as returned by `cooccurrence_counts`, with columns `word1`, `word2`, `a`, `b`, `c`, `d`, `n`.
`alpha`	Significance threshold for classifying a pair as attraction or repulsion rather than neutral. Default 0.05.
`p_adjust_method`	Method passed to `p.adjust` to correct for multiple comparisons across all tested pairs. Default `"BH"` (Benjamini-Hochberg false discovery rate). Set to `"none"` to disable.

Details

The phi coefficient is the Pearson correlation coefficient for two binary variables. For a 2x2 table with cells `a` (both present), `b` (word1 only), `c` (word2 only), `d` (neither), phi is

$$\phi = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$$

Significance is assessed via a chi-squared test on $n\phi^2$, which is asymptotically equivalent to the standard chi-squared test of independence for a 2x2 table (Fisher's exact test is used instead when any cell count is below 5).

Value

cooc with additional columns phi, p_value, p_adjusted, and relationship (one of "attraction", "repulsion", "neutral"), sorted by phi ascending (strongest repulsion first).

Examples

```
df <- data.frame(id = 1:4, text = c("we are honest", "we are fair",
                                    "they are corrupt", "they are selfish"))
tokens <- tokenize_posts(df, text_col = "text", doc_col = "id", stopwords = FALSE)
cooc <- cooccurrence_counts(tokens, min_count = 1)
word_orientation(cooc, alpha = 0.10)
```

Index