Package ‘stylest’

September 16, 2018

Version 0.1.0
Title Estimating Speaker Style Distinctiveness
Depends R (>= 2.10)
Imports corpus, Matrix, stats
Suggests knitr, rmarkdown, testthat, kableExtra
License GPL-3
URL https://github.com/leslie-huang/stylest
BugReports https://github.com/leslie-huang/stylest/issues
LazyData true
Encoding UTF-8
VignetteBuilder knitr, rmarkdown
RoxygenNote 6.0.1.9000
NeedsCompilation no
Author Leslie Huang [aut, cph, cre],
    Patrick O. Perry [aut, cph],
    Arthur Spirling [aut, cph]
Maintainer Leslie Huang <lesliehuang@nyu.edu>
Repository CRAN
Date/Publication 2018-09-16 18:10:03 UTC

R topics documented:

  fit_term_usage .............................................  2
  novels_excerpts ..........................................  2
  print.stylest_model .....................................  3
  stylest ...................................................  4
fit_term_usage  Computes speakers’ term usage rates

Description
Computes speakers’ term usage rates

Usage
fit_term_usage(x, speaker, terms, smooth)

Arguments
- x: Text vector. May be a corpus_frame object
- speaker: Vector of speaker labels. Should be the same length as x
- terms: Vocabulary for document term matrix
- smooth: Numeric value used smooth term frequencies

Value
named list of terms, vector of num tokens uttered by each speaker, smoothing value, and (smoothed) term usage rate matrix

novels_excerpts  Excerpts from English novels

Description
A dataset of text from English novels by Jane Austen, George Eliot, and Elizabeth Gaskell.

Usage
novels_excerpts
Format

A dataframe with 21 rows and 3 variables:

- **title**: Title
- **author**: Author
- **text**: Excerpt of text in complete sentences from the first 1,000 chars of the novel.

Source

Novel excerpts obtained from Project Gutenberg full texts in the public domain in the USA. [http://gutenberg.org](http://gutenberg.org)

---

### print.stylest_model

*Custom print method for stylest_model*

Description

Custom print method for stylest_model

Usage

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

Arguments

- `x`: ‘stylest_model’ object
- `...`: Additional arguments

Value

Prints summary information about the ‘stylest_model’ object

Examples

```r
data(novels_excerpts)
speaker_mod <- stylest_fit(novels_excerpts$text, novels_excerpts$author)
print(speaker_mod)
```
**stylest**

A package for estimating textual distinctiveness

---

### Description

stylest provides a set of functions for fitting a model of speaker distinctiveness, including tools for selecting the optimal vocabulary for the model and predicting the most likely speaker (author) of a new text.

---

### stylest_fit

Fit speaker_model to a corpus

---

#### Description

The main function in stylest, stylest_fit fits a model using a corpus of texts labeled by speaker.

#### Usage

```r
stylest_fit(x, speaker, terms = NULL, filter = NULL, smooth = 0.5)
```

#### Arguments

- `x`: Text vector. May be a `corpus_frame` object
- `speaker`: Vector of speaker labels. Should be the same length as `x`
- `terms`: If not NULL, terms to be used in the model. If NULL, use all terms
- `filter`: If not NULL, a text filter to specify the tokenization. See `corpus` for more information about specifying `filter`
- `smooth`: Numeric value used smooth term frequencies instead of the default of 0.5

#### Details

The user may specify only one of `terms` or `cutoff`. If neither is specified, all terms will be used.

#### Value

A S3 `stylest_model` object containing:
- `speakers`: Vector of unique speakers
- `filter`: text filter used
- `terms`: terms used in fitting the model
- `ntoken`: Vector of number of tokens per speaker
- `smooth`: Smoothing value
- `rate`: Matrix of speaker rates for each term in vocabulary

#### Examples

```r
data(novels_excerpts)
speaker_mod <- stylest_fit(novels_excerpts$text, novels_excerpts$author)
```
**stylest_odds**

**Pairwise prediction of the most likely speaker of texts**

**Description**

Computes the mean log odds of the most likely speaker of each text over pairs of the speaker of a text and every other speaker in the stylest_model.

**Usage**

```r
stylest_odds(model, text, speaker, prior = NULL)
```

**Arguments**

- `model`: stylest_model object
- `text`: Text vector. May be a corpus_frame object
- `speaker`: Vector of speaker labels. Should be the same length as x
- `prior`: Prior probability of speakers. Uses equal prior if NULL

**Value**

A S3 stylest_odds object containing: a stylest_model object; vector of mean log odds that each actual speaker (compared with other speakers in the corpus) spoke their corresponding texts in the corpus; vector of SEs of the log odds

**Examples**

```r
data(novels_excerpts)
speaker_mod <- stylest_fit(novels_excerpts$text, novels_excerpts$author)
stylest_odds(speaker_mod, novels_excerpts$text, novels_excerpts$author)
```

---

**stylest_predict**

**Predict the most likely speaker of a text**

**Description**

Use a fitted stylest_model to predict the most likely speaker of a text. This function may be used on in-sample or out-of-sample texts.

**Usage**

```r
stylest_predict(model, text, prior = NULL)
```
Arguments

- **model** - `stylest_model` object
- **text** - Text vector. May be a `corpus_frame` object
- **prior** - Prior probability, defaults to NULL

Value

- `stylest_predict` object containing:
  - **model** the fitted `stylest_model` object used in prediction,
  - **predicted** the predicted speaker,
  - **log_probs** matrix of log probabilities,
  - **log_prior** matrix of log prior probabilities

Examples

```r
data(novels_excerpts)
speaker_mod <- stylest_fit(novels_excerpts$text, novels_excerpts$author)
stylest_predict(speaker_mod, "This is an example text, who wrote it?")
```

---

**stylest_select_vocab**  
*Select vocabulary using cross-validated out-of-sample prediction*

Description

Selects optimal vocabulary quantile(s) for model fitting using performance on predicting out-of-sample texts.

Usage

```r
stylest_select_vocab(x, speaker, filter = NULL, smooth = 0.5,
                      nfold = 5, cutoff_pcts = c(50, 60, 70, 80, 90, 99))
```

Arguments

- **x** - Corpus as text vector. May be a `corpus_frame` object
- **speaker** - Vector of speaker labels. Should be the same length as x
- **filter** - if not NULL, a `corpus_text_filter`
- **smooth** - value for smoothing. Defaults to 0.5
- **nfold** - Number of folds for cross-validation. Defaults to 5
- **cutoff_pcts** - Vector of cutoff percentages to test. Defaults to c(50, 60, 70, 80, 90, 99)

Value

List of: best cutoff percent with the best speaker classification rate; cutoff percentages that were tested; matrix of the mean percentage of incorrectly identified speakers for each cutoff percent and fold; and the number of folds for cross-validation
Examples

```r
## Not run:
data(novels_excerpts)
stylest_select_vocab(novels_excerpts$text, novels_excerpts$author, cutoff_pcts = c(50, 90))
## End(Not run)
```

---

**stylest_terms**

*Use vocab cutoff to select terms for fitting the model*

**Description**

The same text, speaker, and filter should be used in this model as in `fit_speaker` to select the terms for the latter function.

**Usage**

```r
stylest_terms(x, speaker, vocab_cutoff, filter = NULL)
```

**Arguments**

- `x` : Corpus as text vector. May be a `corpus_frame` object
- `speaker` : Vector of speaker labels. Should be the same length as `x`
- `vocab_cutoff` : Quantile cutoff for the vocabulary in (0, 100]
- `filter` : if not NULL, a corpus filter

**Value**

list of terms

**Examples**

```r
data(novels_excerpts)
stylest_terms(novels_excerpts$text, novels_excerpts$author, vocab_cutoff = 50)
```
stylest_term_influence

*Compute the influence of terms*

**Description**

Compute the influence of terms

**Usage**

```r
stylest_term_influence(model, text, speaker)
```

**Arguments**

- `model`  
  `stylest_model` object
- `text`  
  Text vector. May be a `corpus_frame` object
- `speaker`  
  Vector of speaker labels. Should be the same length as `x`

**Value**

`data.frame` with columns representing terms, their mean influence, and their maximum influence

**Examples**

```r
data(novels_excerpts)
speaker_mod <- stylest_fit(novels_excerpts$text, novels_excerpts$author)
stylest_term_influence(speaker_mod, novels_excerpts$text, novels_excerpts$author)
```
Index

*Topic datasets
  novels_excerpts, 2

fit_term_usage, 2

novels_excerpts, 2

print.stylest_model, 3

stylest, 4
stylest_fit, 4
stylest_odds, 5
stylest_predict, 5
stylest_select_vocab, 6
stylest_term_influence, 8
stylest_terms, 7