Package 'doc2concrete'

June 28, 2022

Type Package

Title Measuring Concreteness in Natural Language

Version 0.5.6

Author Mike Yeomans

Maintainer Mike Yeomans <mk.yeomans@gmail.com>

Description Models for detecting concreteness in natural language. This package is built in support of Yeomans (2021) <doi:10.1016/j.obhdp.2020.10.008>, which reviews linguistic models of concreteness in several domains. Here, we provide an implementation of the best-performing domain-general model (from Brysbaert et al., (2014) <doi:10.3758/s13428-013-0403-5>) as well as two pre-trained models for the feedback and plan-making domains.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports tm, quanteda, parallel, glmnet, stringr, english, textstem, SnowballC, stringi

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-28 19:20:02 UTC

R topics documented:

`adviceNgrams` *Pre-trained advice concreteness features*

Description

For internal use only. This dataset demonstrates the ngram features that are used for the pre-trained adviceModel.

Usage

```
adviceNgrams
```

Format

A (truncated) matrix of ngram feature counts for alignment to the pre-trained advice glmnet model.

Source

Yeomans (2020). A Concrete Application of Open Science for Natural Language Processing.

`bootstrap_list` *Concreteness mTurk Word List*

Description

Word list from Paetzold & Specia (2016). A list of 85,942 words where concreteness was imputed using word embeddings.

Usage

```
bootstrap_list
```

Format

A data frame with 85,942 rows and 2 variables.

Word character text of a word with an entry in this dictionary

Conc.M predicted concreteness score for that word (from 100-700)

Source

#' Paetzold, G., & Specia, L. (2016, June). Inferring psycholinguistic properties of words. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 435-440).

`doc2concrete` *Concreteness Scores*

Description

Detects linguistic markers of concreteness in natural language. This function is the workhorse of the `doc2concrete` package, taking a vector of text documents and returning an equal-length vector of concreteness scores.

Usage

```
doc2concrete(
  texts,
  domain = c("open", "advice", "plans", "finance"),
  wordlist = doc2concrete::mturk_list,
  stop.words = TRUE,
  number.words = TRUE,
  shrink = FALSE,
  fill = FALSE,
  uk_english = FALSE,
  num.mc.cores = 1
)
```

Arguments

`texts`	character A vector of texts, each of which will be tallied for concreteness.
`domain`	character Indicates the domain from which the text data was collected (see details).
`wordlist`	Dictionary to be used. Default is the Brysbaert et al. (2014) list.
`stop.words`	logical Should stop words be kept? Default is TRUE
`number.words`	logical Should numbers be converted to words? Default is TRUE
`shrink`	logical Should open-domain concreteness models regularize low-count words? Default is FALSE.
`fill`	logical Should empty cells be assigned the mean rating? Default is TRUE.
`uk_english`	logical Does the text contain any British English spelling? Including variants (e.g. Canadian). Default is FALSE
`num.mc.cores`	numeric number of cores for parallel processing - see parallel::detectCores(). Default is 1.

Details

In principle, concreteness could be measured from any english text. However, the definition and interpretation of concreteness may vary based on the domain. Here, we provide a domain-specific pre-trained classifier for concreteness in advice & feedback data, which we have empirically confirmed to be robust across a variety of contexts within that domain (Yeomans, 2021).

The training data for the advice classifier includes both second-person (e.g. "You should") and third-person (e.g. "She should") framing, including some names (e.g. "Riley should"). For consistency, we anonymised all our training data to replace any names with "Riley". If you are working with a dataset that includes the names of advice recipients, we recommend you convert all those names to "Riley" as well, to ensure optimal performance of the algorithm (and to respect their privacy).

There are many domains where such pre-training is not yet possible. Accordingly, we provide support for two off-the-shelf concreteness "dictionaries" - i.e. document-level aggregations of word-level scores. We found that that have modest (but consistent) accuracy across domains and contexts. However, we still encourage researchers to train a model of concreteness in their own domain, if possible.

Value

A vector of concreteness scores, with one value for every item in ‘text‘.

References

Yeomans, M. (2021). A Concrete Application of Open Science for Natural Language Processing. Organizational Behavior and Human Decision Processes, 162, 81-94.

Brysbaert, M., Warriner, A. B., & Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. Behavior Research Methods, 46(3), 904-911.

Paetzold, G., & Specia, L. (2016, June). Inferring psycholinguistic properties of words. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 435-440).

Examples

```
data("feedback_dat")

doc2concrete(feedback_dat$feedback, domain="open")


cor(doc2concrete(feedback_dat$feedback, domain="open"),feedback_dat$concrete)
```

feedback_dat *Personal Feedback Dataset*

Description

A dataset containing responses from people on Mechanical Turk, writing feedback to a recent collaborator, that were then scored by other Turkers for feedback specificity. Note that all proper names of advice recipients have been substituted with "Riley" - we recommend the same in your data.

Usage

```
feedback_dat
```

Format

A data frame with 171 rows and 2 variables:

feedback character text of feedback from writers

concrete numeric average specificity score from readers

Source

Blunden, H., Green, P., & Gino, F. (2018).

"The Impersonal Touch: Improving Feedback-Giving with Interpersonal Distance."

Academy of Management Proceedings, 2018.

finance_list *Concreteness Finance Word List*

Description

The mTUrk-annotated list, with the 1,000 most common words re-annotated by finance professionals as an in-domain dictionary.

Usage

```
finance_list
```

Format

A data frame with 40,004 rows and 2 variables.

Word character text of a word with an entry in this dictionary

Conc.M average concreteness score for that word (from 1-5)

Source

Reyt & Yeomans (working paper)

mturk_list *Concreteness mTurk Word List*

Description

Word list from Brysbaert, Warriner & Kuperman (2014). A list of 39,954 words that have been hand-annotated by crowdsourced workers for concreteness.

Usage

```
mturk_list
```

Format

A data frame with 39,954 rows and 2 variables.

Word character text of a word with an entry in this dictionary

Conc.M average concreteness score for that word (from 1-5)

Source

Brysbaert, M., Warriner, A. B., & Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. Behavior Research Methods, 46(3), 904-911.

ngramTokens *Ngram Tokenizer*

Description

Tally bag-of-words ngram features

Usage

```
ngramTokens(
  texts,
  wstem = "all",
  ngrams = 1,
  language = "english",
  punct = TRUE,
  stop.words = TRUE,
  number.words = TRUE,
  overlap = 1,
  sparse = 0.995,
  verbose = FALSE,
  vocabmatch = NULL,
  num.mc.cores = 1
)
```

Arguments

texts	character vector of texts.
wstem	character Which words should be stemmed? Defaults to "all".
ngrams	numeric Vector of ngram lengths to be included. Default is 1 (i.e. unigrams only).
language	Language for stemming. Default is "english"
punct	logical Should punctuation be kept as tokens? Default is TRUE
stop.words	logical Should stop words be kept? Default is TRUE
number.words	logical Should numbers be kept as words? Default is TRUE
overlap	numeric Threshold (as cosine distance) for including ngrams that constitute other included phrases. Default is 1 (i.e. all ngrams included).
sparse	maximum feature sparsity for inclusion (1 = include all features)
verbose	logical Should the package report token counts after each ngram level? Useful for long-running code. Default is FALSE.
vocabmatch	matrix Should the new token count matrix will be coerced to include the same tokens as a previous count matrix? Default is NULL (i.e. no token match).
num.mc.cores	numeric number of cores for parallel processing - see parallel::detectCores(). Default is 1.

Details

This function produces ngram featurizations of text based on the quanteda package. This provides a complement to the doc2concrete function by demonstrating How to build a feature set for training a new detection algorithm in other contexts.

Value

a matrix of feature counts

Examples

```
dim(ngramTokens(feedback_dat$feedback, ngrams=1))
dim(ngramTokens(feedback_dat$feedback, ngrams=1:3))
```

---

planNgrams *Pre-trained plan concreteness features*

Description

For internal use only. This dataset demonstrates the ngram features that are used for the pre-trained planModel.

Usage

```
planNgrams
```

Format

A (truncated) matrix of ngram feature counts for alignment to the pre-trained planning glmnet model.

Source

Yeomans (2020). A Concrete Application of Open Science for Natural Language Processing.

`uk2us` *UK to US Conversion dictionary*

Description

For internal use only. This dataset contains a quanteda dictionary for converting UK words to US words. The models in this package were all trained on US English.

Usage

```
uk2us
```

Format

A quanteda dictionary with named entries. Names are the US version, and entries are the UK version.

Source

Borrowed from the quanteda.dictionaries package on github (from user kbenoit)

Index