Package ‘oolong’

March 21, 2020

Title Create Validation Tests for Automated Content Analysis

Version 0.3.4

Description Intended to create standard human-in-the-loop validity tests for typical automated content analysis such as topic modeling and dictionary-based methods. This package offers a standard workflow with functions to prepare, administer and evaluate a human-in-the-loop validity test. This package provides functions for validating topic models using word intrusion and Topic intrusion tests, as described in Chang et al. (2009) <https://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models>. This package also provides functions for generating gold-standard data which are useful for validating dictionary-based methods. The default settings of all generated tests match those suggested in Chang et al. (2009) and Song et al. (2020) <doi:10.1080/10584609.2020.1723752>.

License LGPL (>= 2.1)

Encoding UTF-8

URL https://github.com/chainsawriot/oolong

LazyData true

Depends R (>= 3.5)

Imports stm, purrr, tibble, shiny, miniUI, text2vec (>= 0.6), digest, R6, quanteda, irr, ggplot2, cowplot, dplyr, stats, utils

RoxygenNote 7.0.2

Suggests testthat (>= 2.1.0), topicmodels, covr, stringr, knitr, rmarkdown

BugReports https://github.com/chainsawriot/oolong/issues

VignetteBuilder knitr

NeedsCompilation no

Author Chung-hong Chan [aut, cre] (<https://orcid.org/0000-0002-6232-7530>)

Maintainer Chung-hong Chan <chainsawtiney@gmail.com>

Repository CRAN

Date/Publication 2020-03-21 17:40:06 UTC

R topics documented:

`abstracts` *Abstracts of communication journals dataset*

Description

This is a random sample of all abstracts of papers published in high-impact communication journals from 2000 to 2017. The one ends with "dfm" is same data in quanteda::dfm (document-feature matrix) format.

Usage

```
abstracts

abstracts_dfm
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2500 rows and 1 columns.

`abstracts_stm` *Topic models trained with the abstracts dataset.*

Description

These are topic models trained with different topic model packages.

Usage

```
abstracts_stm

abstracts_warplda
```

Format

An object of class `STM` of length 11.

`afinn` *AFINN dictionary*

Description

This is the AFINN sentiment dictionary in quanteda::dictionary format.

Usage

```
afinn
```

Format

An object of class `dictionary2` of length 11.

References

Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. arXiv preprint arXiv:1103.2903.

`clone_oolong` *Clone an oolong object*

Description

Clone a new oolong object. The oolong must not be locked and ever coded.

Usage

```
clone_oolong(oolong)
```

Arguments

`oolong`	an oolong object.

Value

an oolong object

Author(s)

Chung-hong Chan

create_oolong *Generate an oolong test*

Description

This function generates an oolong test object that can either be used for validating a topic model or for creating ground truth (gold standard) of a text corpus.

Usage

```
create_oolong(
  input_model = NULL,
  input_corpus = NULL,
  n_top_terms = 5,
  bottom_terms_percentile = 0.6,
  exact_n = NULL,
  frac = 0.01,
  n_top_topics = 3,
  n_topiclabel_words = 8,
  use_frex_words = FALSE,
  difficulty = 1,
  input_dfm = NULL,
  construct = "positive"
)
```

Arguments

`input_model` (gs/tm) a STM, WarpLDA or topicmodels object; if it is NULL, create_oolong assumes that you want to create gold standard.

`input_corpus` (gs/tm) if input_model is not null, it should be the corpus (character vector or quanteda::corpus object) to generate the model object. If input_model and input_corpus are not NULL, topic intrusion test cases are generated. If input_model is null, it generates gold standard test cases.

`n_top_terms` (tm) integer, number of top topic words to be included in the candidates of word intrusion test.

`bottom_terms_percentile` (tm) double, a term is considered to be an word intruder when its theta less than the percentile of this theta, must be within the range of 0 to 1

`exact_n` (tm/gs) integer, number of topic intrusion test cases to generate, ignore if frac is not NULL

`frac` (tm/gs) double, fraction of test cases to be generated from the corpus

`n_top_topics` (tm) integer, number of most relevant topics to be shown alongside the intruder topic

`n_topiclabel_words` (tm) integer, number of topic words to be shown as the topic label `use_frex_words` (tm) logical, for a STM object, use FREX words if TRUE, use PROB words if FALSE

`difficulty` (tm) double, adjust the difficulty of the test. Higher value indicates higher difficulty and must be within the range of 0 to 1, no effect for STM if use_frex_words is FALSE. Ignore for topicmodels objects.

`input_dfm` (tm) a dfm object used for training the input_model, if input_model is a WarpLDA object

`construct` (gs) string, an adjective to describe the construct you want your coders to code the the gold standard test cases.

Value

an oolong test object.

Usage

This function generates an oolong test object based on `input_model` and `input_corpus`. If `input_model` is not NULL, it generates oolong test for a topic model (tm). If `input_model` is NULL but input_corpus is not NULL, it generates oolong test for generating gold standard (gs).

Methods

An oolong object, depends on its purpose, has the following methods:

`$do_word_intrusion_test()` (tm) launch the shiny-based word intrusion test. The coder should find out the intruder word that is not related to other words.

`$do_topic_intrusion_test()` (tm) launch the shiny-based topic intrusion test. The coder should find out the intruder topic that is least likely to be the topic of the document.

`$do_gold_standard_test()` (gs) launch the shiny-based test for generating gold standard. The coder should determine the level of the predetermined constructs with a 5-point Likert scale.

`$lock(force = FALSE)` (gs/tm) lock the object so that it cannot be changed anymore. It enables `summarize_oolong` and the following method.

`$turn_gold()` (gs) convert the oolong object into a quanteda compatible corpus.

For more details, please see the overview vignette: `vignette("overview",package = "oolong")`

Author(s)

Chung-hong Chan

References

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Advances in neural information processing systems (pp. 288-296).

Song et al. (2020) In validations we trust? The impact of imperfect human annotations as a gold standard on the quality of validation of automated content analysis. Political Communication.

Examples

```
## Creation of oolong test with only word intrusion test
data(abstracts_stm)
data(abstracts)
oolong_test <- create_oolong(input_model = abstracts_stm)
## Creation of oolong test with both word intrusion test and topic intrusion test
oolong_test <- create_oolong(input_model = abstracts_stm, input_corpus = abstracts$text)
## Creation of gold standard
oolong_test <- create_oolong(input_corpus = trump2k)
```

oolong — *Oolong: create and administrate validation tests for typical automated content analysis tools*

Description

The oolong package makes it very easy to create, administrate and evaluate typical automated content analysis tools by providing a framework for creating human-in-the-loop validation tests. For topic models, oolong can generate tests such as word intrusion test and topic intrusion test (Chang et al.). For dictionary-based methods, oolong can generate standardized interface for making gold standard ('Ground truth') data. There are only two core functions of this package: `create_oolong` and `summarize_oolong`.

Author(s)

Chung-hong Chan

print.oolong_gold_standard — *Print oolong gold standard object*

Description

This function prints a summary of the oolong gold standard object. An oolong gold standard object is a result of $turn_gold() method. It is a quanteda::corpus compatible object.

Usage

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

Arguments

x	an oolong gold standard object
...	other parameters

Value

None, a summary of the quanteda::corpus and what you should do are displayed

Author(s)

Chung-hong Chan

`print.oolong_summary` *Print and plot oolong summary*

Description

These functions print or plot a useful summary of the results from `summarize_oolong`. For details, please see the overview vignette: `vignette("overview",package = "oolong")`

Usage

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

## S3 method for class 'oolong_summary'
plot(x, ...)
```

Arguments

`x`	an oolong_summary
`...`	other parameters

Value

None

Summary

Print function displays the following information:

Mean model precision (tm) Higher value indicates better topic interpretability

Quantiles of model precision (tm) Higher value indicates better topic interpretability

P-value of the model precision (tm) Model precision's p-value calculated by one-sample binomial test and Fisher's Omnibus method.

Krippendorff's alpha (gs/tm) Krippendorff's Alpha, if more than one oolong object is analyzed.

K Precision (tm) Model precision for each topic.

Mean TLO (tm) Mean topic log odds, higher value indicates better interpretability

Median TLO (tm) Median topic log odds, higher value indicates better interpretability

Quantiles of TLO (tm) Quantiles of topic log odds

P-Value of the median TLO (tm) Median topic log odds's p-value calculated by permutation test.

Correlation (average answer) (gs) Pearson's correlation between average answer and target value

Corrlation (content length) (gs) Pearson's correlation between content length and target value

Diagnostic plot

Plot function displays a diagnostic plot with the following subplots (gs only).

Top left Correlation between answer from coders and target value to check for correlation between two values. Both axes are minmax transformed.

Top right Bland-altman plot of answer from coders and target value to check for agreement between two values.

Bottom left Correlation between target value and content length to check for the influence of content length.

Bottom right Cook's distance to check for influential observations.

Author(s)

Chung-hong Chan

`summarise_oolong` *Summarize oolong objects*

Description

This function summarizes one or more oolong objects. All oolong objects must be locked.

Usage

```
summarise_oolong(..., target_value = NULL)

summarize_oolong(..., target_value = NULL)
```

Arguments

`...`	(tm/gs) one or more oolong objects to be summarized.
`target_value`	(gs) a vector of numeric values, the value you want to validate against the human-coded gold standard. One example of this target value is sentiment score extracted automatically from text.

Value

An oolong summary. Depends on purpose, an oolong summary object has the following values:

`$type` (gs/tm) type of analysis, either 'gs' or 'tm'

`$kripp_aplha` (gs/tm) Krippendorff's Alpha, if more than one oolong object is analyzed.

`$rater_precision` (tm) Model precision

`$res$rater_precision_p_value` (tm) Model precision's p-value calculated by one-sample binomial test and Fisher's Omnibus method.

`$k_precision` (tm) precision for each topic

`$tlo` (tm) vector of topic log odds

`$tlo_pvalue` (tm) Median topic log odds's p-value calculated by permutation test.

`$cor` (gs) Pearson's correlation between average answer and target value

`$cor_length` (gs) Pearson's correlation between content length and target value

`$diag_plot` (gs) diagnostic plot.

A useful summary of an object can be obtained either by `print.oolong_summary` or `plot.oolong_summary`. For details, please see the overview vignette: `vignette("overview",package = "oolong")`

Author(s)

Chung-hong Chan

References

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Advances in neural information processing systems (pp. 288-296).

Song et al. (2020) In validations we trust? The impact of imperfect human annotations as a gold standard on the quality of validation of automated content analysis. Political Communication.

Examples

```
# Please try this example in interactive R sessions only.
if (interactive()) {
   data(abstracts_stm)
   oolong_test1 <- create_oolong(abstracts_stm)
   oolong_test2 <- clone_oolong(oolong_test1)
   oolong_test1$do_word_intrusion_test()
   oolong_test2$do_word_intrusion_test()
   oolong_test1$lock()
   oolong_test2$lock()
   summarize_oolong(oolong_test1, oolong_test2)
}
```

trump2k *Trump's tweets dataset*

Description

This is a random sample of 2000 tweets from @realdonaldtrump account before his assumption of duty as the president of the United States.

Usage

```
trump2k
```

Format

An object of class `character` of length 2000.

Index