The Journal of Open Source Software

oolong: An R package for validating automated content analysis tools

Chung-hong Chan[1] **and Marius Sältzer**[1]

1 Mannheimer Zentrum für Europäische Sozialforschung, Universität Mannheim

DOI:

Software

- Review
- Repository
- Archive

Submitted:
Published:

Summary

Oolong is an R package providing functions for validating common automated content analysis techniques such as topic modeling and dictionary-based methods. This package is designed for R users needing to validate these methods. Typical users of oolong are communication and political scientists, among others.

Validity is a requirement of content analysis (Krippendorff, 2018; Neuendorf, 2016). Validation of automated methods has been called for by many scholars, e.g. Grimmer & Stewart (2013); Ribeiro, Araújo, Gonçalves, Gonçalves, & Benevenuto (2016); Van Atteveldt & Peng (2018). However, the method for validating these automated methods is unstandardized.

Oolong makes it easy to generate standard validation tests suggested by Chang, Gerrish, Wang, Boyd-Graber, & Blei (2009) and Song et al. (2020).

Validating topic models

Topic models can be validated by word intrusion test and topic intrusion test (Chang et al., 2009). In these tests, a human rater is asked to pick an odd word from a bunch of words (word intrusion test) or pick an odd topic from a bunch of topics for a document (topic intrusion test). Oolong provides an easy-to-use Shiny interface for these tests (Figure 1).

Currently, oolong supports a variety of topic models, e.g. structural topic models / correlated topic models from `stm` (Roberts, Stewart, & Tingley, 2019), warp-LDA models from `text2vec` (Selivanov, Bickel, & Wang, 2020), latent dirichlet allocation / correlated-topic models from `topicmodels` (Grün & Hornik, 2011), biterm topic models from `BTM` (Wijffels, 2020) and keyword-assisted topic models from `keyATM` (Eshima, Imai, & Sasaki, 2020).

For instance, `abstracts_stm` is a structural topic model trained with the text data from `abstracts$text` (Chan & Grill, 2020).

```
library(stm)
library(tibble)
library(dplyr)
library(quanteda)
library(oolong)
```

The Journal of Open Source Software

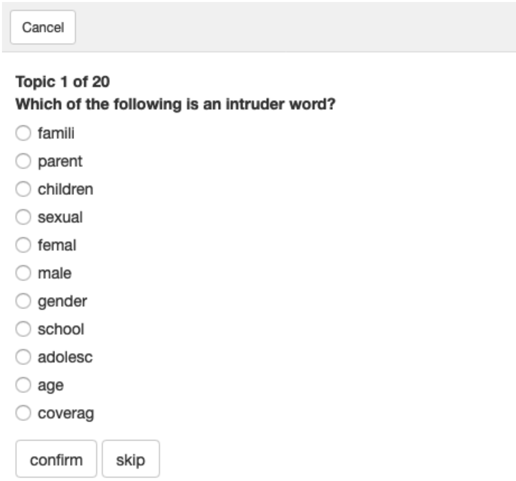

Figure 1: A screenshot of word intrusion test

```r
abstracts_stm
```

```
## A topic model with 20 topics, 2500 documents and a 3998 word dictionary.
```

```r
abstracts
```

```
## # A tibble: 2,500 x 1
##    text
##    <chr>
##  1 This study explores the benefits and risks featured in medical tourism broke
##  2 This article puts forth the argument that with the transfer of stock trading
##  3 The purpose of this study was to evaluate the effect the visual fidelity of
##  4 Among the many health issues relevant to college students, overconsumption o
##  5 This address, delivered at ICA's 50th anniversary conference, calls on the a
##  6 The Internet has often been used to reach men who have sex with men (MSMs) i
##  7 This article argues that the literature describing the internet revolution i
##  8 This research study examined Bud Goodall's online health narrative as a case
##  9 Information technology and new media allow for collecting and sharing person
## 10 Using a national, telephone survey of 1,762 adolescents aged 12-17 years, th
## # ... with 2,490 more rows
```

The function `create_oolong` creates a test object with both word intrusion test and topic intrusion test.

```r
oolong_test <- create_oolong(input_model = abstracts_stm,
                             input_corpus = abstracts$text)
oolong_test
```

```
## An oolong test object with k = 20, 0 coded.
## Use the method $do_word_intrusion_test() to do word intrusion test.
## With 25 cases of topic intrusion test. 0 coded.
## Use the method $do_topic_intrusion_test() to do topic intrusion test.
## Use the method $lock() to finalize this object and see the results.
```

The tests can be administered with methods do_word_intrusion_test and do_topic_intrusion_test.

```
oolong_test$do_word_intrusion_test()
oolong_test$do_topic_intrusion_test()
```

After both tests has been done by a human rater, the test object must be locked and then accuracy metrics such as model precision (MP) and TLO (topic log odd) are displayed.

```
oolong_test$lock()
oolong_test
```

```
## An oolong test object with k = 20, 20 coded.
## 95%  precision
## With 25 cases of topic intrusion test. 25 coded.
## TLO: -0.079
```

The suggested workflow is to have at least two human raters to do the same set of tests. Test object can be cloned to allow multiple raters to do the test. More than one test object can be studied together using the function summarize_oolong().

```
oolong_test_rater1 <- create_oolong(abstracts_stm, abstracts$text)
oolong_test_rater2 <- clone_oolong(oolong_test_rater1)
```

```
## Let rater 1 do the test.
oolong_test_rater1$do_word_intrusion_test()
oolong_test_rater1$do_topic_intrusion_test()
oolong_test_rater1$lock()

## Let rater 2 do the test.
oolong_test_rater2$do_word_intrusion_test()
oolong_test_rater2$do_topic_intrusion_test()
oolong_test_rater2$lock()
```

Get a summary of the two objects.

```
summarize_oolong(oolong_test_rater1, oolong_test_rater2)
```

```
## New names:
## * NA -> ...1
## * NA -> ...2
```

```
## Mean model precision: 0.4
## Quantiles of model precision: 0.35, 0.375, 0.4, 0.425, 0.45
## P-value of the model precision (H0: Model precision is not better than random g
## Krippendorff's alpha: 0.1875
## K Precision: 0.5, 0, 0, 0, 0, 1, 0, 0, 0.5, 0, 0.5, 1, 0.5, 0.5, 0.5, 0.5, 1, 1
## Mean TLO: -1.88
## Median TLO: -0.65
## Quantiles of TLO: -6.35936400839407, -3.71810391983537, -0.64508286087911, 0, 0
## P-Value of the median TLO (H0: Median TLO is not better than random guess): 0
```

Validating dictionary-based methods

Dictionary-based methods such as AFINN (Nielsen, 2011) can be validated by creating a gold standard dataset (Song et al., 2020). Oolong provides a workflow for generating such gold standard dataset.

For example, you are interested in studying the sentiment of tweets from Donald Trump. `trump2k` is a random subset of 2,000 tweets from Donald Trump. And you would like to use AFINN to extract sentiment from these tweets. In this analysis, AFINN sentiment score is the *target value*.

```
tibble(text = trump2k)
```

```
## # A tibble: 2,000 x 1
##    text
##    <chr>
##  1 "In just out book, Secret Service Agent Gary Byrne doesn't believe that Croo
##  2 "Hillary Clinton has announced that she is letting her husband out to campai
##  3 "\"@TheBrodyFile: Always great to visit with @TheBrodyFile one-on-one with \
##  4 "Explain to @brithume and @megynkelly, who know nothing, that I will beat Hi
##  5 "Nobody beats me on National Security. https://t.co/sCrj4Ha1I5"
##  6 "\"@realbill2016: @realDonaldTrump @Brainykid2010 @shl Trump leading LA Time
##  7 "\"@teapartynews: Trump Wins Tea Party Group's 'Nashville Straw Poll' - News
##  8 "Big Republican Dinner tonight at Mar-a-Lago in Palm Beach. I will be there!
##  9 ".@HillaryClinton loves to lie. America has had enough of the CLINTON'S! It
## 10 "\"@brianstoya: @realDonaldTrump For POTUS #2016\""
## # ... with 1,990 more rows
```

A test object can be generated also with `create_oolong`. The argument `construct` should be an adjective, e.g. “positive” or “liberal”.

```
trump <- create_oolong(input_corpus = trump2k,
                       construct = "positive",
                       exact_n = 20)
trump
```

```
## An oolong test object (gold standard generation) with 20 cases, 0 coded.
## Use the method $do_gold_standard_test() to generate gold standard.
## Use the method $lock() to finalize this object and see the results.
```

Similarly, we suggest to have at least two human coders to do the same set of tests.

```
trump2 <- clone_oolong(trump)
```

Instruct two coders to code the tweets and lock the objects.

```
trump$do_gold_standard_test()
trump2$do_gold_standard_test()
trump$lock()
trump2$lock()
```

The method `turn_gold` converts a test object into a quanteda corpus (Benoit et al., 2018).

```r
gold_standard <- trump$turn_gold()
gold_standard
```

```
## Corpus consisting of 20 documents and 1 docvar.
## text1 :
## "Thank you Eau Claire, Wisconsin.  #VoteTrump on Tuesday, Apr..."
##
## text2 :
## ""@bobby990r_1: @realDonaldTrump would lead polls the second ..."
##
## text3 :
## ""@KdanielsK: @misstcassidy @AllAboutTheTea_ @realDonaldTrump..."
##
## text4 :
## "Thank you for a great afternoon Birmingham, Alabama! #Trump2..."
##
## text5 :
## ""@THETAINTEDT: @foxandfriends @realDonaldTrump Trump 2016 ht..."
##
## text6 :
## "People believe CNN these days almost as little as they belie..."
##
## [ reached max_ndoc ... 14 more documents ]
## Access the answer from the coding with quanteda::docvars(obj, 'answer')
```

This corpus can be used to calculate the target value, e.g. AFINN.

```r
dfm(gold_standard, remove_punct = TRUE) %>% dfm_lookup(afinn) %>%
    quanteda::convert(to = "data.frame") %>%
    mutate(matching_word_valence = (neg5 * -5) + (neg4 * -4) +
               (neg3 * -3) + (neg2 * -2) + (neg1 * -1) +
               (zero * 0) + (pos1 * 1) + (pos2 * 2) + (pos3 * 3) +
               (pos4 * 4) + (pos5 * 5),
           base = ntoken(gold_standard, remove_punct = TRUE),
           afinn_score = matching_word_valence / base) %>%
           pull(afinn_score) -> afinn_score
```

Summarize all oolong objects with the target value.

```r
res <- summarize_oolong(trump, trump2, target_value = afinn_score)
```

Printing the summary shows Krippendorff's Alpha, an indicator of interrater reliability. The validity metrics of a text analytic method can be tinted by poor interrater reliability of manual annotations (Song et al., 2020). It is important to ensure high interrater reliability first.

```r
res
```

```
## Krippendorff's Alpha: 0.931443661971831
## Correlation: 0.744 (p = 0)
## Effect of content length: -0.323 (p = 0.164)
```

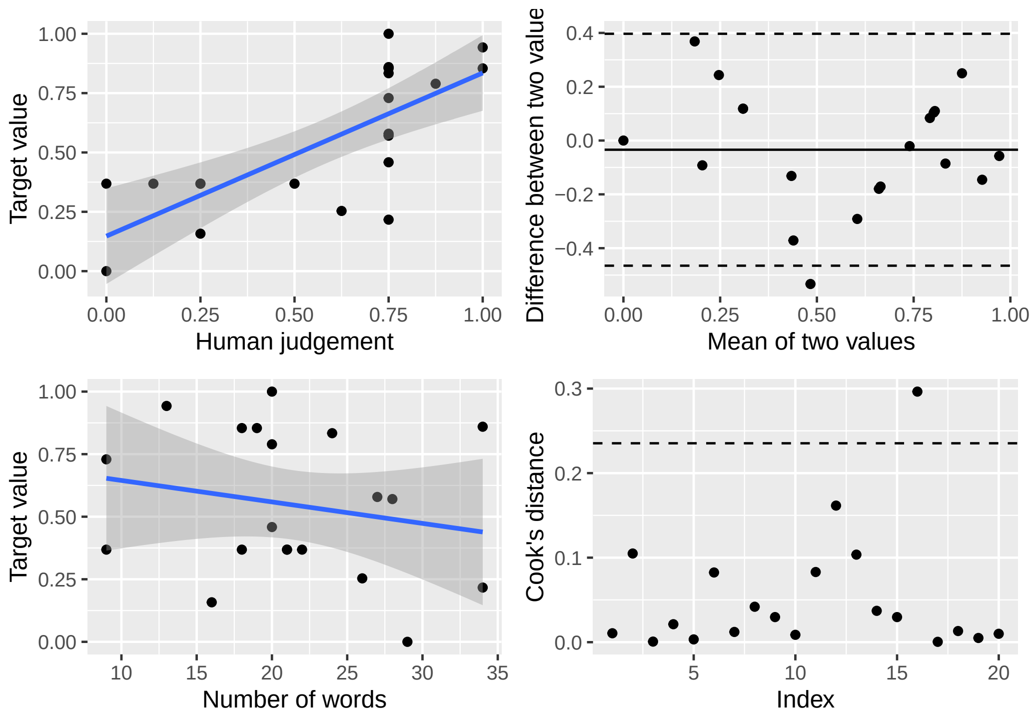

Figure 2: Diagnostic plots generated by oolong

Additional diagnostic plots can also be displayed.

```
plot(res)
```

The 4 subplots from left to right, top to bottom are: 1) correlation between human judgement and target value; 2) Bland-Altman plot; 3) correlation between target value and content length and 4) Cook's distance of all data point. These plots are helpful to determine criterion validity, agreement, robustness against content length and outliers of the target value.

Acknowledgements

The development of oolong is partially supported by SAGE Concept Grant.

References

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). Quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, *3*(30), 774. doi:10.21105/joss.00774

Chan, C.-h., & Grill, C. (2020). The highs in communication research: Research topics with high supply, high popularity and high prestige in high-impact journals. *Communication Research*, *forthcoming*.

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 informa-*

tion processing systems (pp. 288–296). Retrieved from https://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models

Eshima, S., Imai, K., & Sasaki, T. (2020). Keyword assisted topic models. *arXiv preprint arXiv:2004.05964*.

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*, *21*(3), 267–297. doi:10.1093/pan/mps028

Grün, B., & Hornik, K. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30. doi:10.18637/jss.v040.i13

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. SAGE.

Neuendorf, K. A. (2016). *The content analysis guidebook*. SAGE.

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

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, *5*(1), 1–29. doi:10.1140/epjds/s13688-016-0085-1

Roberts, M. E., Stewart, B. M., & Tingley, D. (2019). stm: An R package for structural topic models. *Journal of Statistical Software*, *91*(2), 1–40. doi:10.18637/jss.v091.i02

Selivanov, D., Bickel, M., & Wang, Q. (2020). *Text2vec: Modern text mining framework for R*. Retrieved from https://CRAN.R-project.org/package=text2vec

Song, H., Tolochko, P., Eberl, J.-M., Eisele, O., Greussing, E., Heidenreich, T., Lind, F., 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*, 1–23. doi:10.1080/10584609.2020.1723752

Van Atteveldt, W., & Peng, T.-Q. (2018). When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science. *Communication Methods and Measures*, *12*(2-3), 81–92. doi:10.1080/19312458.2018.1458084

Wijffels, J. (2020). *BTM: Biterm topic models for short text*. Retrieved from https://CRAN.R-project.org/package=BTM