TopLing

Evaluating the Quality of Machine Translation for Multilingual Topic Modeling

Nadezhda (Nadja) Ozornina   Yannik Peters   Mario Haim

2025-11-28

At a glance

This tutorial provides guidelines on evaluating the quality of machine translation as a consolidation strategy for multilingual topic modeling in `R`.

By the end of this tutorial, you will be able to:

- Apply machine translation for text consolidation.
- Perform topic modeling on translated data.
- Measure metrics to evaluate the quality of machine translation for topic modeling.
- Identify topics with potential translation issues and inspect them.

Table of Content

1. Introduction

Background

Topic modeling is a widely used text analysis approach in computational social science, enabling researchers to identify latent thematic structures of large-scale text collections (Jacobi, Van Atteveldt, and Welbers 2016). However, the outcomes of this method are highly sensitive to data quality and preprocessing decisions (see Peters and Shah 2024), making consistent text preparation essential for producing interpretable and comparable results.

A particularly important scenario for assessing data quality arises when working with text collections in multiple languages (Lind et al. 2022). In such cases, machine translation is often applied as a consolidation strategy (Reber 2019), helping to bring multilingual data to a common denominator before the analysis. When combined with probabilistic topic modeling, this approach enables the identification of cross-lingual topics (Chan et al. 2020), however, the reliability of findings can be compromised by errors introduced during translation. This dependence on input quality represents a significant challenge in computational text analysis (Baden et al. 2022) and highlights the need for clear guidelines on evaluating machine translation outputs and understanding their impact on topic modeling results.

Aims and Scope

In this tutorial, we provide practical guidelines for evaluating machine translation quality for multilingual topic modeling, using a United Nations corpus in German language (Eisele and Chen 2010). The documents ($N = 994$) are consolidated into English using *Google Translate*, and then translated back into German. On original German texts and texts resulted from back-and-forth translation, the algorithms of structural topic modeling (STM, Roberts, Stewart, and Tingley (2019)) are applied. The results are compared using metrics of (a) feature overlap, (b) topical prevalence, and (c) topical content (De Vries, Schoonvelde, and Schumacher 2018), with outputs available for qualitative examination. Finally, we discuss the strengths and limitations of this approach in the context of computational social science.

2. Setup

First, we load all relevant libraries. Please ensure to have all packages installed using the `install.packages()` command. The `piercer` package is only available from *GitHub*, so you need to use the `remotes` package to install it.

```
# Processing
library(tidyverse)

# Translation
library(polyglotr)

# Text analysis
library(quanteda)
library(stm)

# Plotting
library(VennDiagram)
library(grid)

# Calculations
library(proxy)
library(remotes)
library(gtools)

if (!requireNamespace("piercer", quietly = TRUE)) {
    remotes::install_github("sjpierce/piercer")
}

library(piercer)
```

Second, we load the United Nations documents in German, representing a subsample of the open-source multilingual corpus originally published by Eisele and Chen (2010). The dataset includes resolutions and related documents and is used here as an example for evaluating translation quality. When applying this workflow to your own multilingual projects, you can use the provided scripts on subsamples in any of the languages included in your data.

```
# Open and inspect the dataset
documents <- readRDS("documents.rds")
glimpse(documents)
```

```
Rows: 994
Columns: 2
$ id       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
…
$ original <chr> "Resolution 1752 (2007) verabschiedet auf der 5661. Sitzung d
…
```

```
cat("Mean document length is", mean(str_count(documents$original, "\\S+")))
```

```
Mean document length is 1146.676
```

We see that our sample contains 994 documents in German, each consisting of slightly more than 1,000 words on average. This number of documents was chosen to illustrate the applicability of the tool while maintaining feasibility for analysis.

3. Tool application

3.1. Translation

The next step is to translate the German documents into English, being the *lingua franca* for consolidating multilingual data (De Vries, Schoonvelde, and Schumacher 2018), and then back into German. This process is called back-and-forth translation and is commonly used to evaluate the quality of machine translation systems (e.g. Baden and Stalpouskaya 2015). The more the topic model derived from the original German texts aligns with the model from the back-translated texts, the higher is the translation quality.

For that, we first define a function which interacts with interface of *Google Translate* in the background and translates texts from a given language into English. The function reports any errors, discards affected documents from the dataset, and prints their IDs for later inspection.

```
# Define the translation function
translate <- function(docs_to_translate, lang_original, lang_translate, doc_ids) {

  # Create a function for one document
  translate_document <- function(doc, i) {
    tryCatch({
      res <- google_translate_long_text(doc,
                                        source = lang_original,
                                        target = lang_translate,
                                        chunk_size = 1000,
                                        preserve_newlines = FALSE)
      message("Document ", i, " translated from ", lang_original, " into ", lang_translate)
      res
    }, error = function(e) {
      message("Document ", i, " translation failed: ", e$message)
      NA_character_
    })
  }

  # Apply the function to all documents in the sample
  translated <- Vectorize(translate_document,
```

```
                          vectorize.args = c("doc", "i"))(
    docs_to_translate, doc_ids)

  # Create a tibble and remove failed translations
  tib <- tibble(id = doc_ids,
    original = docs_to_translate,
    translated = translated)
  tib <- tib[!is.na(tib$translated), ]
  tib
}
```

We apply the function to translate German texts into English and then back into German.

> **Important**
>
> Due to interacting directly with *Google Translate*, the provided algorithm runs very slow: in our case, translating all documents took about one hour per language pair. To speed up the translation process, we recommend using API services from Google or DeepL (for comparison between tools, see Reber (2019)). To bypass the speed limitations and enable direct use of the tool, we provide ready-made translations collected in November 2025 in the next section (3.2).

```
# Translate documents from German into English
#translation_forth <- translate(documents$text, "de", "en", documents$id)

# Translate documents from English back into German
#translation_back <- translate(translation_forth$translated, "en", "de", #translation_forth$

# Save translated data
#saveRDS(translation_forth, "translation_forth.rds")
#saveRDS(translation_back, "translation_back.rds")
```

3.2. Preprocessing

First, we read and inspect the translated data obtained after applying *Google Translate* to the analyzed documents. We also load a comprehensive list of German stopwords, combining spacyr (Benoit and Matsuo 2017) and quanteda (Benoit et al. 2018), extended with Roman numerals relevant to our study. Other word lists or domain-specific stopwords can be applied as needed.

```
# Read in translated data and stopwords
translation_forth <- readRDS("translation_forth.rds")
translation_back <- readRDS("translation_back.rds")
stopwords <- readRDS("stopwords.rds")

# Show data
cat("translation_forth:\n")
```

```
translation_forth:
```

```
glimpse(translation_forth)
```

```
Rows: 994
Columns: 3
$ id         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
…
$ original   <chr> "Resolution 1752 (2007) verabschiedet auf der 5661. Sitzung
…
$ translated <chr> "Resolution 1752 (2007) adopted at the 5661st session of th
…
```

```
cat("translation_back:\n")
```

```
translation_back:
```

```
glimpse(translation_back)
```

```
Rows: 994
Columns: 3
$ id         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
…
$ original   <chr> "Resolution 1752 (2007) adopted at the 5661st session of th
…
$ translated <chr> "Resolution 1752 (2007), angenommen auf der 5661. Sitzung d
…
```

```
cat("stopwords:\n")
```

```
stopwords:
```

```
glimpse(stopwords)
```

```
Rows: 617
Columns: 1
$ var <chr> "beim", "derjenigen", "die", "dich", "jahren", "muss", "wessen", "
…
```

Here, the dataframe `translation_forth` contains the results of translating German texts into English, with the column `original` for German texts and `translated` for their English versions. The dataframe `translation_back` contains the results of translating these English texts back into German, with the column `original` holding the English texts and `translated` containing the final results in German.

Next, we define a preprocessing function that performs tokenization, stopword removal, stemming with `quanteda`, and relative pruning. For this, we follow the preprocessing guidelines from Maier et al. (2018) and Van Atteveldt, Trilling, and Calderón (2022). For real-world projects, we recommend applying lemmatization with `spacyr` instead of stemming, which is used here due to its simpler installation requirements.

```r
# Create preprocessing function
process_and_create_dfm <- function(translation_df, field, stopwords) {

  # Create corpus
  corpus <- translation_df %>%
    corpus(text_field = field)

  # Tokenize and preprocess
  tokens_clean <- corpus %>%
    tokens(what = "word",
           remove_punct = TRUE,
           remove_symbols = TRUE,
           remove_numbers = TRUE,
           remove_url = TRUE,
           remove_separators = TRUE,
           split_hyphens = TRUE,
           split_tags = TRUE,
           include_docvars = FALSE) %>%
    tokens_tolower() %>%
    tokens_select(stopwords,
                  selection = "remove") %>%
    tokens_wordstem()

  # Create DFM and apply relative pruning
  dfm_result <- dfm(tokens_clean)
  dfm_result <- dfm_trim(dfm_result,
                         min_docfreq = 0.005,
                         max_docfreq = 0.99,
                         docfreq_type = 'prop',
                         verbose = TRUE)

  return(dfm_result)
}
```

We apply this function to the original German texts as well as to the results of translating these texts back and forth into German. Next, we examine the document-feature matrices (DFMs) generated through preprocessing.

```r
# Apply preprocessing
dfm_original <- process_and_create_dfm(translation_forth, "original", stopwords) # original
dfm_translated <- process_and_create_dfm(translation_back, "translated", stopwords) # transl

# Print preprocessing summary
cat("DFM from Original Texts (dfm_original):\n")
```

```
DFM from Original Texts (dfm_original):
```

```r
dfm_original
```

```
Document-feature matrix of: 994 documents, 7,236 features (95.54% sparse) and 0 docvars.
       features
```

```
docs    resolut verabschiedet auf sitzung sicherheitsrat am april unter hinwei
  text1       3             1  20       1              5  3     2     6      2
  text2       2             1  16       1             21  7     1     6      0
  text3       4             1   3       1              4  2     0     0      0
  text4       3             0   6       0              1  0     1     0      0
  text5       4             1  18       0              1  1     1     4      3
  text6       6             0   4       0              1  0     0     2      2
       features
docs    all
  text1   5
  text2   5
  text3   1
  text4   1
  text5   2
  text6   0
[ reached max_ndoc ... 988 more documents, reached max_nfeat ... 7,226 more features ]
```

```
cat("DFM after Back-and-forth Translation (dfm_translated):\n")
```

```
DFM after Back-and-forth Translation (dfm_translated):
```

```
dfm_translated
```

```
Document-feature matrix of: 994 documents, 7,046 features (95.46% sparse) and 0 docvars.
       features
docs    resolut angenommen auf sitzung sicherheitsrat am april erinnert an all
  text1       4          1  21       1              5  2     2        2  5   7
  text2       2          0  10       3             21  7     1        0  1   5
  text3       4          1   5       1              4  2     0        0  0   1
  text4       2          0   4       0              1  0     1        0  3   1
  text5       3          1  21       0              1  1     1        0  3   2
  text6       5          0   4       1              1  0     0        0  3   0
[ reached max_ndoc ... 988 more documents, reached max_nfeat ... 7,036 more features ]
```

```
# Save the results
saveRDS(dfm_original, file = "dfm_original.rds")
saveRDS(dfm_translated, file = "dfm_translated.rds")
```

With preprocessing complete, we can move to topic modeling.

3.3. Topic Modeling

3.3.1. Choosing the number of topics

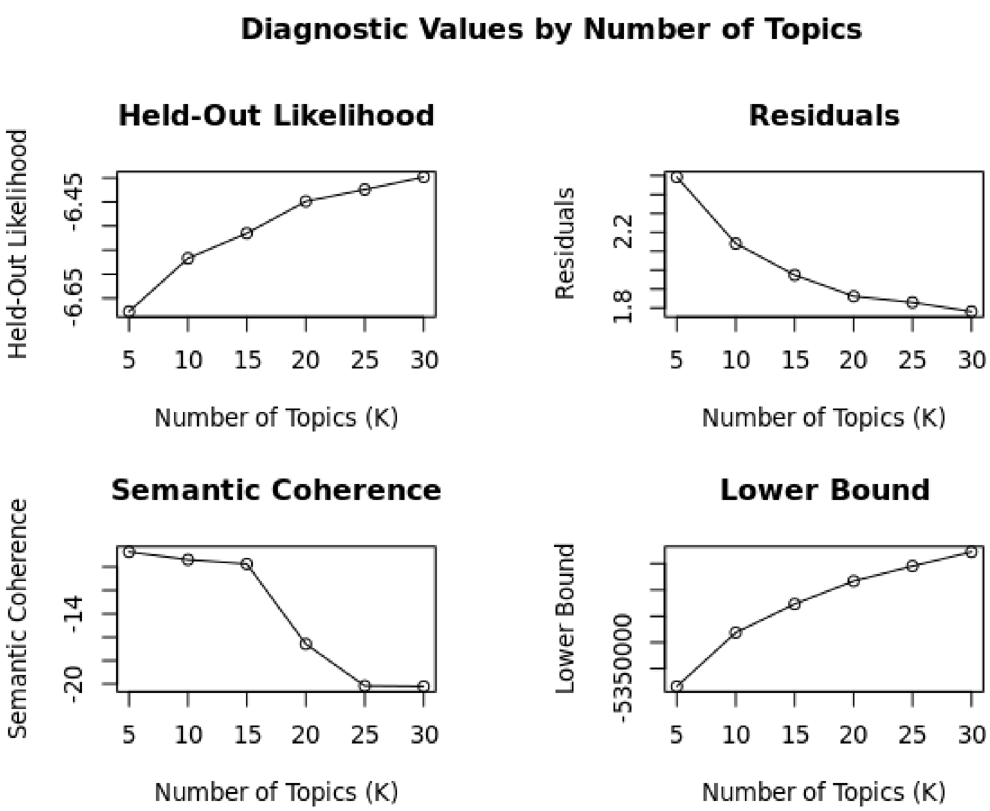
Now, we want to apply topic modeling separate to the DFM `dfm_original`, based on original German texts, and to the DFM `dfm_translated`, obtained from translating the texts to English and back into German.

The first step in topic modeling is to determine the optimal number of topics (K) for both analyzed DFMs. Since the focus of this tutorial is on examining the machine translation quality, we use a simplified approach for selecting the number of topics, utilizing the built-in functionality in the STM package (Roberts, Stewart, and Tingley 2019). For more robust analyses, it is strongly recommended to consider multiple evaluation metrics and qualitative assessments; for guidance, see Maier et al. (2018) and Bernhard, Teuffenbach, and Boomgaarden (2023).

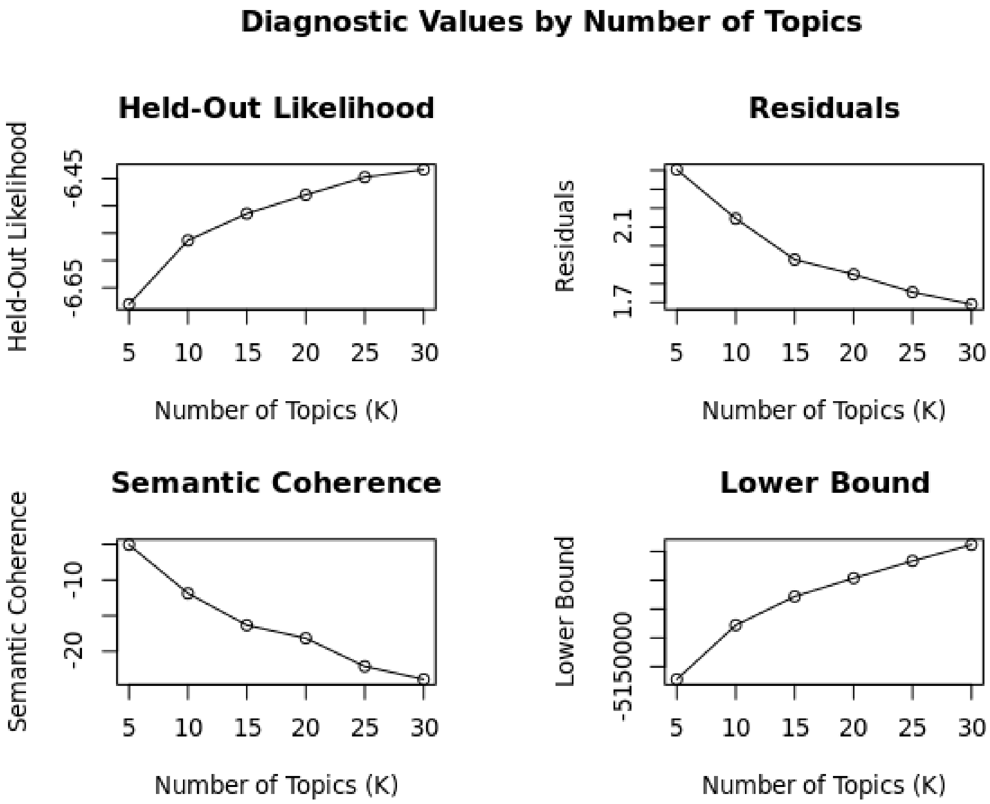
```r
# Important: Execution time may vary by hardware and can take up to approximately 10 minutes
# Prepare the data and define the numbers of topics to be examined
dfm_original_stm <- dfm_original %>%
  convert(to = "stm")
dfm_translated_stm <- dfm_translated %>%
  convert(to = "stm")
Ks <- seq(5, 30, 5) # test for 5, 10, 15, 20, 25, 30 topics

# Apply function for metrics calculation from the STM package
kresult_original <- searchK(documents = dfm_original_stm$documents,
                            vocab = dfm_original_stm$vocab,
                            K = Ks,
                            init.type = "Spectral",
                            verbose = FALSE)
kresult_translated <- searchK(documents = dfm_translated_stm$documents,
                              vocab = dfm_translated_stm$vocab,
                              K = Ks,
                              init.type = "Spectral",
                              verbose = FALSE)

# Plot the results
plot(kresult_original, main = "Number of Topics (K) for Original Texts")
```



```r
plot(kresult_translated, main = "Number of Topics (K) for Back-and-forth Translation")
```



We then evaluate the plots to identify the solution that produces the most coherent topics, with the lowest residuals and the highest held-out likelihood (for details, see Roberts, Stewart, and Tingley (2019)). Based on these metrics, the optimal number of topics appears to be between 10 and 15 for both DFMs. Accordingly, we choose to fit a model with 10 topics.

3.3.2. Calculating the topic models

The second step is to estimate the topic models themselves. For that, we use the STM package and apply spectral initialization to ensure the reproducibility of the outcomes. We then print a summary of the fitted model along with the top words for each topic. Here, FREX words are recommended for labeling and qualitative examination, as they balance frequency and exclusivity, making them well-suited to characterize each topic.

```r
# Important: Execution time may vary by hardware and can take up to approximately 5 minutes.
# Calculate topic models
model_original <- stm(documents = dfm_original_stm$documents,
                      vocab = dfm_original_stm$vocab,
                      K = 10,
                      init.type = "Spectral",
                      max.em.its = 500,
                      verbose = FALSE)
model_translated <- stm(documents = dfm_translated_stm$documents,
                        vocab = dfm_translated_stm$vocab,
                        K = 10,
                        init.type = "Spectral",
```

```
                        max.em.its = 500,
                        verbose = FALSE)

# Print model summary
cat("Summary of Model for Original Texts (model_original):\n")
```

```
Summary of Model for Original Texts (model_original):
```

```
summary(model_original)
```

```
A topic model with 10 topics, 994 documents and a 7236 word dictionary.

Topic 1 Top Words:
     Highest Prob: den, für, bericht, generalsekretär, generalversammlung, vom, zu
     FREX: zweijahreszeitraum, aufsichtsdienst, programmhaushaltsplan, rechnungsprüf, inspek
     Lift: einnahmenkapitel, beitragsausschuss, 7a, anlageausschuss, arbeitssprachen, aufzur
     Score: haushaltsfragen, beratenden, aufsichtsdienst, rechnungsprüf, zweijahreszeitraum,
Topic 2 Top Words:
     Highest Prob: zu, den, auf, in, von, für, dem
     FREX: weltraum, verbrechensverhütung, strafrechtspfleg, terrorismus, seerechtsübereinko
     Lift: anwendungsmöglichkeiten, himmelskörp, mond, neunundvierzigst, raumfahrtnationen,
     Score: weltraum, technik, seerechtsübereinkommen, strafrechtspfleg, unterausschuss, ver
Topic 3 Top Words:
     Highest Prob: in, zu, den, mit, sicherheitsrat, auf, von
     FREX: somalia, afghanistan, republik, monuc, afghanischen, liberia, kongo
     Lift: goma, grenzkommiss, lina, marcoussi, n'djamena, zurückhaltung, ivorischen
     Score: monuc, kongo, unifil, afghanischen, afghanistan, unoci, somalia
Topic 4 Top Words:
     Highest Prob: auf, in, zu, den, dass, von, dem
     FREX: palästinensischen, besetzten, selbstregierung, jerusalem, völker, israel, hilfswe
     Lift: arab, befreiungsorganis, beraubt, declar, durchführungsabkommen, endow, fremder
     Score: palästinensischen, jerusalem, israel, golan, besetzten, menschenrecht, israelisc
Topic 5 Top Words:
     Highest Prob: den, in, für, von, vom, dollar, zu
     FREX: betrag, dollar, personalabgab, höhe, geschätzten, anzurechnen, versorgungsbasi
     Lift: sonderhaushalt, anteilmäßig, anzurechnen, ausgeschöpften, ausrüstungsgegenstände
     Score: dollar, betrag, personalabgab, anzurechnen, geschätzten, guthaben, mission
Topic 6 Top Words:
     Highest Prob: den, in, zu, auf, für, vereinten, von
     FREX: kultur, dekad, süd, aktionsprogramm, partnerschaft, entwicklungsfinanzierung, ent
     Lift: bezeichnen, einkommensdisparitäten, einzelziel, friedens2, katastrophenvorsorg, k
     Score: dekad, süd, entwicklungsfinanzierung, entwicklung, kultur, nachhaltig, aktionspr
Topic 7 Top Words:
     Highest Prob: in, zu, ein, den, für, von, auf
     FREX: wir, ich, unser, rechtsstaatlichkeit, vertragsorgan, un, press
     Lift: grundsatzpolitischen, informationsausschuss, marktzugang, armen, unser, wir, akad
     Score: wir, ich, unser, rechtsstaatlichkeit, vertragsorgan, press, armen
Topic 8 Top Words:
     Highest Prob: oder, in, zu, ein, von, den, mit
     FREX: vertragsstaat, richter, irak, absatz, artikel, sei, handlung
     Lift: annimmt, auszulegen, eingefroren, einzufrieren, erdöl, höchstzahl, hypotheken
```

```
     Score: vertragsstaat, richter, straftat, handlung, vertragsstaaten, habe, litem
Topic 9 Top Words:
     Highest Prob: zu, von, in, den, auf, dass, über
     FREX: folter, kindern, mädchen, bewaffneten, kinder, kambodscha, kind
     Lift: auszuhandelnden, einschüchterungshandlungen, fakultativprotokol, getretenen, grun
     Score: bewaffneten, mädchen, folter, kindern, konflikten, staatennachfolg, konflikt
Topic 10 Top Words:
     Highest Prob: von, in, auf, zu, den, vom, über
     FREX: kernwaffen, abrüstung, nichtverbreitung, vertrag, abrüstungskonferenz, nuklearen,
     Lift: chemisch, kernwaffenfrei, rüstungskontroll, abrüstungskommiss, abrüstungsübereink
     Score: kernwaffen, chemisch, vertrag, gc, nichtverbreitung, abrüstung, kernwaffenstaate
```

```r
cat("Summary of Model for Back-and-forth Translation (model_translated):\n")
```

```
Summary of Model for Back-and-forth Translation (model_translated):
```

```r
summary(model_translated)
```

```
A topic model with 10 topics, 994 documents and a 7046 word dictionary.
```

```
Topic 1 Top Words:
     Highest Prob: vom, dezemb, generalversammlung, resolut, a, den, vereinten
     FREX: jugoslawien, sprachversionen, dezemb, überarbeitung, redaktionel, generalausschus
     Lift: wohnort, beitragsausschuss, ausgabenlast, komoren, vollmachtenausschuss, xix, sei
     Score: vollmachtenausschuss, plenarsitzung, tagesordnungspunkt, dezemb, jugoslawien, ta
Topic 2 Top Words:
     Highest Prob: oder, ein, von, in, zu, den, für
     FREX: vertragsstaat, richter, innerstaatlichen, litem, artikel, handlung, sei
     Lift: angab, dieselb, eingefroren, einreichen, einziehung, einzufrieren, gegenständ
     Score: vertragsstaat, richter, vertragsstaaten, rent, artikel, litem, straftat
Topic 3 Top Words:
     Highest Prob: in, zu, den, sicherheitsrat, auf, von, für
     FREX: republik, kongo, monuc, liberia, demokratischen, parteien, côte
     Lift: abchasien, abchasisch, abgezogen, ami, beigeordnetem, burundischen, d’
ivoir
     Score: monuc, kongo, unifil, unoci, darfur, liberia, friedensabkommen
Topic 4 Top Words:
     Highest Prob: zu, auf, in, von, den, dass, menschenrecht
     FREX: besetzten, palästinensischen, rassismus, fremdenfeindlichkeit, grundfreiheiten, i
     Lift: annexion, befreiungsorganis, dreiundzwanzigsten, durchführungsvereinbarungen, end
     Score: palästinensischen, menschenrecht, rassendiskriminierung, recht, rassismus, fremd
Topic 5 Top Words:
     Highest Prob: zu, in, für, ein, den, vereinten, von
     FREX: armut, partnerschaft, wenigsten, entwickelten, süd, entwicklungsziel, unser
     Lift: armen, ausgebaut, benachteiligt, brücken, einkommensunterschied, empfängerländern
     Score: armut, süd, entwicklungsfinanzierung, rechtsstaatlichkeit, wenigsten, unser, akt
Topic 6 Top Words:
     Highest Prob: den, für, in, von, vom, dollar, us
     FREX: betrag, us, dollar, höhe, versorgungsbasi, einnahmen, finanzzeitraum
     Lift: ausgeschöpften, barzahlungen, beitragstabell, beschaffungskosten, bewilligten, fe
     Score: dollar, us, betrag, finanzzeitraum, endenden, personalabgaben, sonderhaushalt
```

```
Topic 7 Top Words:
     Highest Prob: von, zu, in, auf, den, zur, über
     FREX: kernwaffen, nichtverbreitung, abrüstung, nuklearen, vertrag, atomwaffen, zone
     Lift: achtundvierzigst, aktivitätenprogramm, angrenzend, atomtest, atomwaffenfreien, at
     Score: kernwaffen, chemisch, nichtverbreitung, nuklearen, gc, atomwaffen, vertrag
Topic 8 Top Words:
     Highest Prob: zu, auf, den, in, von, zur, mit
     FREX: weltraum, kriminalprävent, strafjustiz, seerechtsübereinkommen, neukaledonien, ko
     Lift: einundvierzigsten, raumfahrtnationen, straftätern, überprüfungsbestimmungen, welt
     Score: weltraum, neukaledonien, staatennachfolg, seerechtsübereinkommen, strafjustiz, k
Topic 9 Top Words:
     Highest Prob: zu, in, von, auf, den, für, vereinten
     FREX: afghanistan, afghanischen, humanitären, bewaffneten, konflikten, humanitär, kinde
     Lift: afghanen, bodengestützt, drogenkontrollstrategi, eupol, frauengruppen, herkunftsc
     Score: afghanischen, afghanistan, bewaffneten, afghanisch, konflikten, humanitären, kon
Topic 10 Top Words:
     Highest Prob: den, zu, für, in, generalsekretär, bericht, auf
     FREX: aufsichtsdienst, programmhaushalt, amtssprachen, handelsrecht, press, konferenzau
     Lift: konferenzausschuss, arbeitssprachen, auslastung, berichts1, bewertungsmethoden, c
     Score: beratenden, haushaltsfragen, aufsichtsdienst, zweijahresperiod, press, dienstort
```

```r
# Save the results
saveRDS(model_original, file = "model_original.rds")
saveRDS(model_translated, file = "model_translated.rds")
```

3.3.3. Topic matching and comparison

To assess whether topic modeling produced similar topics for the original German texts and the back-and-forth translated texts, we match the topics from both models (`model_original` and `model_translated`) based on similarities in their word probabilities. This approach allows us to identify the most similar (matching) topics across the two models. The function, originally based on the approach by De Vries, Schoonvelde, and Schumacher (2018), also extracts and pre-saves values comparing topic distributions and top words, which we use in the subsequent analysis steps for comparison (3.4).

```r
# Load topic models
model_original <- readRDS("model_original.rds")
model_translated <- readRDS("model_translated.rds")

# Create function for topic matching and comparison
comparizer <- function(topics, model_original, model_translated) {

  # Extract terms
  model_original.terms <- t(labelTopics(model_original, n = 50)$frex)
  colnames(model_original.terms) <- paste("Topic", 1:ncol(model_original.terms))

  model_translated.terms <- t(labelTopics(model_translated, n = 50)$frex)
  colnames(model_translated.terms) <- paste("Topic", 1:ncol(model_translated.terms))

  # Extract topic probabilities
  model_original.topicprobs <- as.matrix(model_original$theta)
  model_translated.topicprobs <- as.matrix(model_translated$theta)
```

```r
# Extract word probabilities
model_original.wordprobs <- t(exp(model_original$beta$logbeta[[1]]))
rownames(model_original.wordprobs) <- model_original$vocab
colnames(model_original.wordprobs) <- 1:ncol(model_original.wordprobs)
model_translated.wordprobs <- t(exp(model_translated$beta$logbeta[[1]]))
rownames(model_translated.wordprobs) <- model_translated$vocab
colnames(model_translated.wordprobs) <- 1:ncol(model_translated.wordprobs)

# Identify overlapping words
words_overlap <- intersect(rownames(model_original.wordprobs), rownames(model_translated.w

# Keep only overlapping words, alphabetically ordered
original_wordprobs_temp <- model_original.wordprobs[words_overlap, , drop = FALSE] %>%
  as_tibble(rownames = "word") %>%
  arrange(word) %>%
  column_to_rownames("word")
translated_wordprobs_temp <- model_translated.wordprobs[words_overlap, , drop = FALSE] %>%
  as_tibble(rownames = "word") %>%
  arrange(word) %>%
  column_to_rownames("word")

# Compute topic pairs for each word
topic_comb <- lapply(rownames(original_wordprobs_temp), function(word) {
  paste(
    which.max(original_wordprobs_temp[word, ]),
    which.max(translated_wordprobs_temp[word, ]),
    sep = ","
  )
})

# Count and sort topic pairs
topic_comb_tab <- sort(table(unlist(topic_comb)), decreasing = TRUE)
topic_comb_names <- names(topic_comb_tab)

# Create unique topic pairs
topic_original <- character()
topic_translated <- character()
topic_unique <- unlist(lapply(topic_comb_names, function(tc) {
  temp <- strsplit(tc, ",", fixed = TRUE)[[1]]
  original <- temp[1]
  translated <- temp[2]
  if (!original %in% topic_original && !translated %in% topic_translated) {
    topic_original <<- c(topic_original, original)
    topic_translated <<- c(topic_translated, translated)
    paste(original, ",", translated, sep = "")
  }
}))

# Identify unassigned topics
not_assigned_original <- setdiff(as.character(seq_len(topics)), topic_original)
not_assigned_translated <- setdiff(as.character(seq_len(topics)), topic_translated)
```

```
topic_unique <- mixedsort(topic_unique)
print("Unique topic pairs:")
print(topic_unique)
print("Not assigned topics in original model:")
print(not_assigned_original)
print("Not assigned topics in translated model:")
print(not_assigned_translated)

# Compute translated column order and reorder matrices
translated_order <- as.numeric(sapply(topic_unique, function(tu) strsplit(tu, ",")[[1]][2]
original_order <- c(not_assigned_original, translated_order)
colnames(model_original.wordprobs) <- colnames(model_original.topicprobs) <- colnames(mode
colnames(model_translated.wordprobs) <- colnames(model_translated.topicprobs) <- colnames(
if(length(not_assigned_original) > 0){
  remove_idx <- as.numeric(not_assigned_original)
  model_original.wordprobs <- model_original.wordprobs[, -remove_idx]
  model_original.topicprobs <- model_original.topicprobs[, -remove_idx]
  model_original.terms <- model_original.terms[, -remove_idx]
  original_wordprobs_temp <- original_wordprobs_temp[, -remove_idx]
}
model_translated.wordprobs <- model_translated.wordprobs[, translated_order]
model_translated.topicprobs <- model_translated.topicprobs[, translated_order]
model_translated.terms <- model_translated.terms[, translated_order]
translated_wordprobs_temp <- translated_wordprobs_temp[, translated_order]
if(length(not_assigned_translated) > 0){
  remove_idx <- seq_len(length(not_assigned_translated))
  model_translated.wordprobs <- model_translated.wordprobs[, -remove_idx]
  model_translated.topicprobs <- model_translated.topicprobs[, -remove_idx]
  model_translated.terms <- model_translated.terms[, -remove_idx]
  translated_wordprobs_temp <- translated_wordprobs_temp[, -remove_idx]
}

# Compute similarity matrices
doc2DocDistrCor <- simil(model_original.topicprobs, model_translated.topicprobs, method="c
doc2DocDistrCos <- simil(model_original.topicprobs, model_translated.topicprobs, method="c
topic2TopicDistrCor <- simil(model_original.topicprobs, model_translated.topicprobs, metho
topic2TopicDistrCos <- simil(model_original.topicprobs, model_translated.topicprobs, metho
topic2TopicSimilCor <- simil(original_wordprobs_temp, translated_wordprobs_temp, method="c
topic2TopicSimilCos <- simil(original_wordprobs_temp, translated_wordprobs_temp, method="c

# Return all results as a list
return(list(
  model_original_terms = model_original.terms,
  model_translated_terms = model_translated.terms,
  model_original_topicprobs = model_original.topicprobs,
  model_translated_topicprobs = model_translated.topicprobs,
  model_original_wordprobs = model_original.wordprobs,
  model_translated_wordprobs = model_translated.wordprobs,
  unique_pairs = topic_unique,
  doc2DocDistrCor = doc2DocDistrCor,
  doc2DocDistrCos = doc2DocDistrCos,
  topic2TopicDistrCor = topic2TopicDistrCor,
```

```r
    topic2TopicDistrCos = topic2TopicDistrCos,
    topic2TopicSimilCor = topic2TopicSimilCor,
    topic2TopicSimilCos = topic2TopicSimilCos
  ))
}

# Apply
results <- comparizer(10, model_original, model_translated)
```

```
[1] "Unique topic pairs:"
 [1] "1,10" "2,8"  "3,3"  "4,4"  "5,6"  "6,1"  "7,5"  "8,2"  "9,9"  "10,7"
[1] "Not assigned topics in original model:"
character(0)
[1] "Not assigned topics in translated model:"
character(0)
```

```r
saveRDS(results, "results.rds")
```

We observe that all topics in our models were successfully matched. If this is not the case for your data, we recommend examining the unmatched topics to identify potential reasons for discrepancies, including issues related to machine translation.

3.4. Evaluating translation quality

3.4.1. Feature overlap

We begin with the first metric for evaluating the quality of machine translation, called *feature overlap* (De Vries, Schoonvelde, and Schumacher 2018). This metric measures the extent to which the DFMs use the same features (in our case, stems), with a higher degree of overlap indicating stronger alignment. The results are expressed as a percentage, reflecting how much the translated data share a similar vocabulary with the original texts.

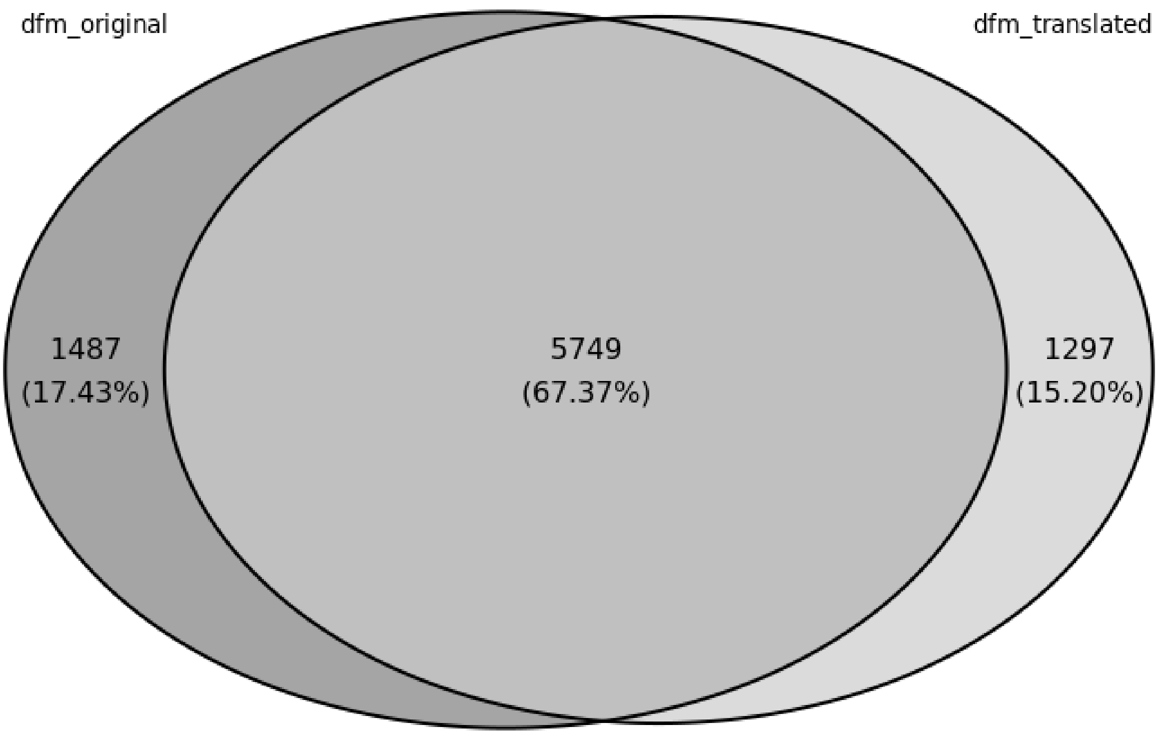
```r
# Load DFMs
dfm_original <- readRDS("dfm_original.rds")
dfm_translated <- readRDS("dfm_translated.rds")

# Calculate feature counts
features_original <- length(unique(featnames(dfm_original)))
features_translated <- length(unique(featnames(dfm_translated)))
features_overlap <- length(intersect(featnames(dfm_original), featnames(dfm_translated)))
features_total <- features_original + features_translated - features_overlap

# Draw Venn diagram with percentages
venn <- draw.pairwise.venn(area1 = features_original,
                           area2 = features_translated,
                           cross.area = features_overlap,
                           cex = 1.0,
                           cat.cex = 0.8,
                           fill = c("gray50", "gray80"),
                           alpha = c(0.7, 0.7),
                           lty = "solid",
```

```
                             col = "black",
                             cat.fontfamily = "Helvetica",
                             fontfamily = "Helvetica",
                             ind = FALSE
                             )
venn_labels <- sapply(venn[5:7], function(x) {
  label <- as.numeric(x$label)
  sprintf("%d\n(%.2f%%)", label, label / features_total * 100)
})
for (i in 5:7) venn[[i]]$label <- venn_labels[i - 4]

# Draw diagram
grid.newpage()
grid.draw(venn)
grid.text("dfm_original", 0.1, 0.9, gp = gpar(fontsize = 10))
grid.text("dfm_translated", 0.9, 0.9, gp = gpar(fontsize = 10))
```



The output is a Venn diagram showing that 67.37% of features from the two DFMs appear in both the original and translated data. This generally aligns with previous studies on the impact of machine translation on topic models (De Vries, Schoonvelde, and Schumacher 2018; Reber 2019), which report slightly higher overlaps (around 75%) for parallel corpora. For other language pairs and text types, overlap benchmarks may be lower, for example, to around 50% in journalistic texts.

3.4.2. Topical prevalence

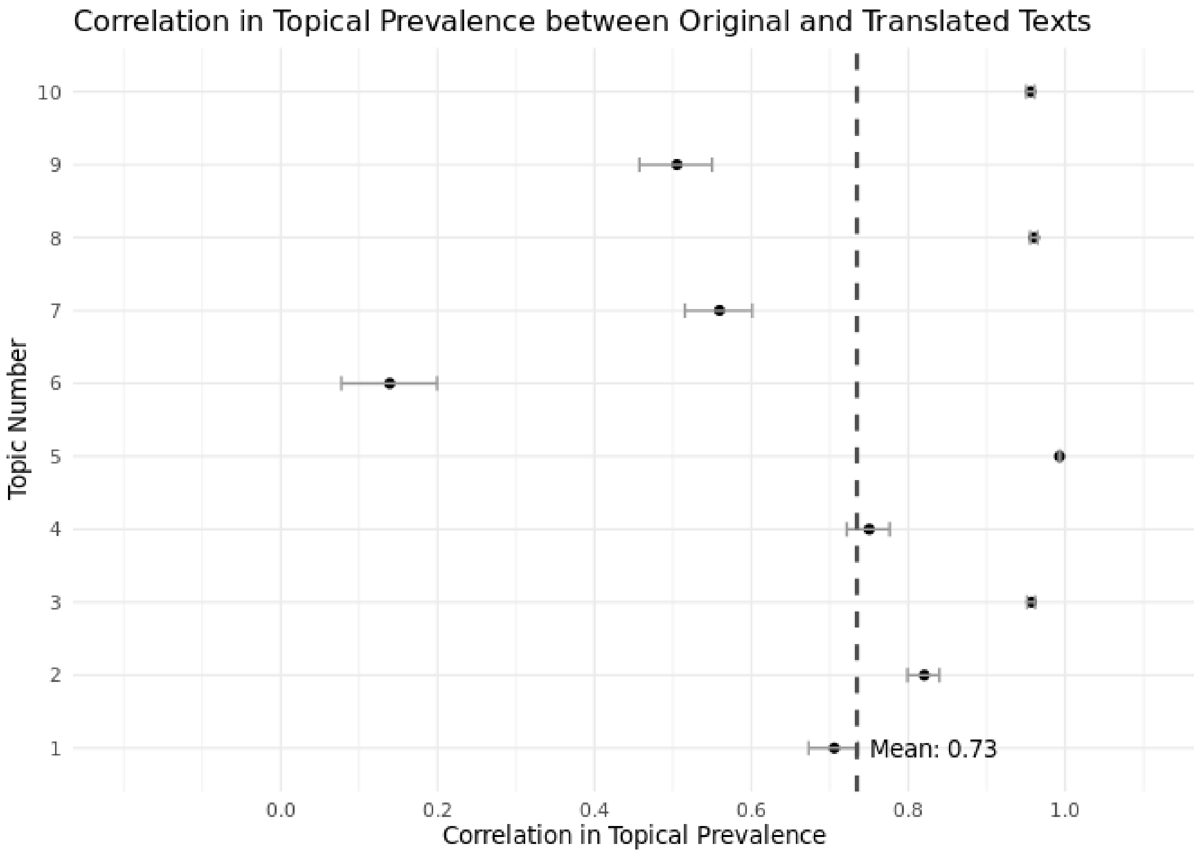
The subsequent comparisons are conducted at the level of the topic models. The next metric, called *topical prevalence*, indicates the extent to which the matched topics are similarly distributed across the original and translated data. This is assessed through correlations computed for each topic pair and averaged across all topics. The algorithms for these calculations are detailed in De Vries, Schoonvelde, and Schumacher (2018).

```
# Load data
model_original <- readRDS("model_original.rds")
model_translated <- readRDS("model_translated.rds")
results <- readRDS("results.rds")

# Create function for calculating confidence intervals
compute_mean_statistics <- function(data, ndoc) {
  ci <- ci.rp(data, ndoc)
  tibble(id = seq_along(data),
         corr = data,
         topic_id = factor(seq_along(data), levels = seq_along(data)),
         overall_mean = mean(data),
         ci.CI.LL = ci$CI.LL,
         ci.CI.UL = ci$CI.UL) %>%
    arrange(corr)
  }

# Create function for topical prevalence
create_prevalence_plot <- function(data, title_text) {
  ggplot(data,
         aes(x = corr, y = topic_id)) +
    geom_point(size = 1.5) +
    geom_errorbar(aes(xmin = ci.CI.LL, xmax = ci.CI.UL),
                  width = 0.2,
                  color = "gray60",
                  size = 0.5) +
    geom_vline(xintercept = mean(data$corr),
               linetype = "dashed",
               color = "gray30",
               size = 0.8) +
    annotate("text",
             x = mean(data$corr),
             y = 1,
             label = paste0("Mean: ", round(mean(data$corr), 2)),
             hjust = -0.1,
             size = 3.5,
             color = "black") +
    labs(title = title_text,
         x = "Correlation in Topical Prevalence",
         y = "Topic Number") +
    scale_x_continuous(limits = c(-0.2, 1.1),
                       breaks = seq(0, 1, 0.2)) +
    theme_minimal(base_size = 10)
}
```

```r
# Apply to the data
topic2TopicDistrCor <- compute_mean_statistics(as.vector(results$topic2TopicDistrCor),
                                               nrow(model_original$theta))
create_prevalence_plot(topic2TopicDistrCor,
  "Correlation in Topical Prevalence between Original and Translated Texts")
```



On average, the matched topics show a similarity of 0.73 in their distribution across the original and translated corpora, indicating sufficient agreement in topical prevalence. This aligns with previous findings, where De Vries, Schoonvelde, and Schumacher (2018) and Reber (2019) report correlations of around 0.70 for parallel texts. We also observe that topic pair 6, where the number refers to the topic in the original German model, has a noticeably lower correlation, which may be attributed to potential translation issues.

In addition to this evaluation, it is also possible to examine topical prevalence at the document level. We do not include this analysis here, as it produces nearly identical results. For examples of such calculations, see the supplementary materials of De Vries, Schoonvelde, and Schumacher (2018).

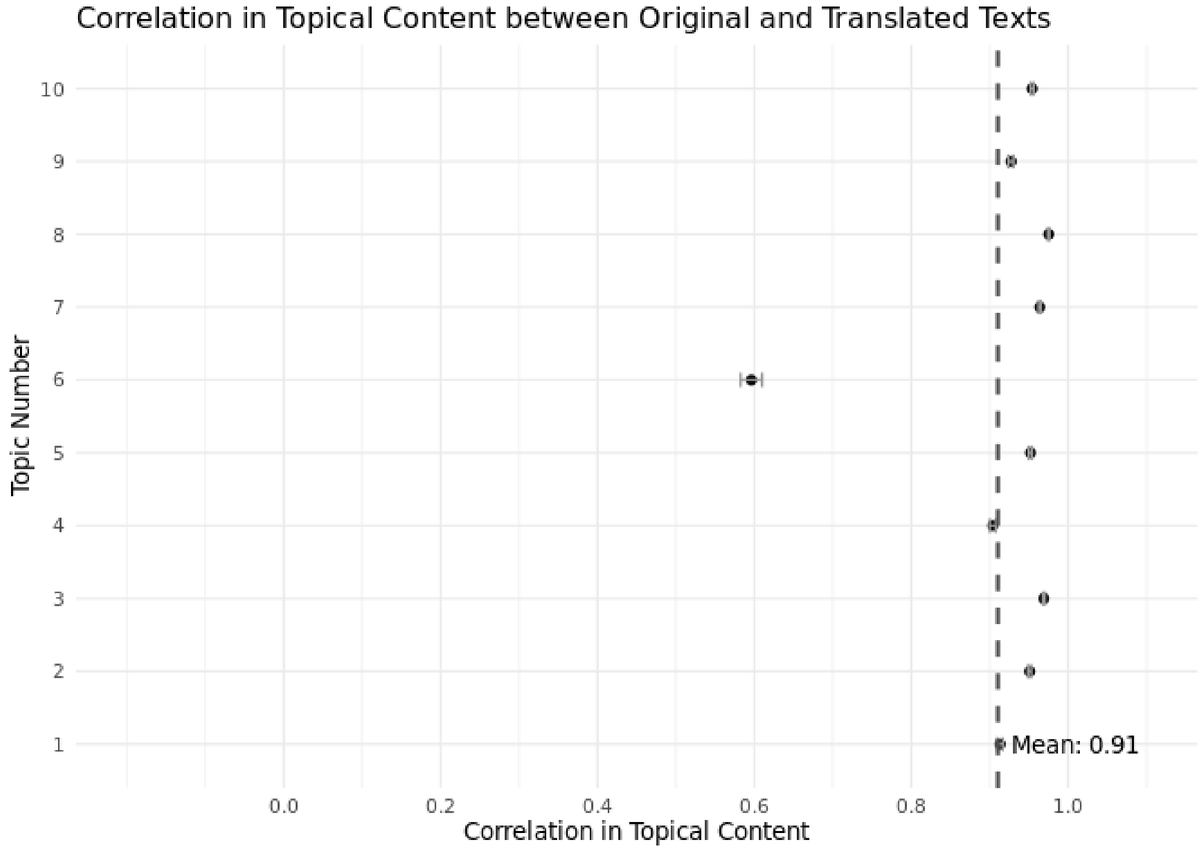
3.4.3. Topical content

Topical content indicates the extent to which the word distributions across matched topics from the compared models correlate with each other. Higher correlations reflect greater similarity in the word distributions within topics and show stronger alignment in the underlying vocabulary of the topic matches.

```r
# Load data
results <- readRDS("results.rds")
```

```r
# Create function for topical content
create_content_plot <- function(data, title_text) {
  mean_corr <- mean(data$corr)
  ggplot(data,
         aes(x = corr, y = topic_id)) +
    geom_point(size = 1.5,
               color = "black") +
    geom_errorbar(aes(xmin = ci.CI.LL, xmax = ci.CI.UL),
                  width = 0.2,
                  color = "gray60",
                  size = 0.5) +
    geom_vline(xintercept = mean_corr,
               linetype = "dashed",
               color = "gray30",
               size = 0.8) +
    annotate("text",
             x = mean_corr, y = 1,
             label = sprintf("Mean: %.2f", mean_corr),
             hjust = -0.1,
             size = 3.5,
             color = "black") +
    labs(title = title_text,
         x = "Correlation in Topical Content",
         y = "Topic Number") +
    scale_x_continuous(limits = c(-0.2, 1.1),
                       breaks = seq(0, 1, 0.2)) +
    theme_minimal(base_size = 10)
}

# Apply to the data
# Important: for this function, steps 3.4.1. and 3.4.2. have to be executed
topic2TopicSimilCor <- compute_mean_statistics(as.vector(results$topic2TopicSimilCor),
                                               features_total)
create_content_plot(topic2TopicSimilCor,
  "Correlation in Topical Content between Original and Translated Texts")
```



The results indicate that word distributions across matched topics are highly similar, with an average correlation of 0.91, higher values around 0.70 reported in De Vries, Schoonvelde, and Schumacher (2018). As with topical prevalence, all topic pairs except for topic 6 show strong similarity. This outlier will be examined qualitatively in the next step of the analysis.

3.4. Strategies for qualitative assessment

The final step of the analysis is to qualitatively evaluate the possible reasons for mismatches in topical prevalence and topical content between the original and translated data. In this step, we can extract the top words and documents for the topic pairs identified as relevant in the previous section. Based on this evaluation, we chose to investigate the inconsistencies in topic pair number 6.

```
# Load data
model_original <- readRDS("model_original.rds")
model_translated <- readRDS("model_translated.rds")
results <- readRDS("results.rds")
translation_forth <- readRDS("translation_forth.rds")
translation_back <- readRDS("translation_back.rds")

# Create function for extraction of top words and documents
inspect_topic_pairs <- function(model_original_terms, model_translated_terms,
                                model_original_theta, model_translated_theta,
                                unique_pairs, topic_pair_number,
                                texts_original, texts_translated,
                                top_n_words, top_n_docs) {
  # Split topic numbers
  topic_original = topic_pair_number
```

```r
  match_pair <- unique_pairs[grepl(paste0("^", topic_original, ","), unique_pairs)]
  if (length(match_pair) == 0) {
    stop("No matching translated topic found for original topic ", topic_original)
  }
  topic_translated <- as.numeric(strsplit(match_pair, ",")[[1]][2])
  cat("Comparing topic pair: Original =", topic_original, ", Translated =", topic_translated

  # Top words
  cat("\nTop words (original):\n")
  print(model_original_terms[1:top_n_words, topic_original])
  cat("\nTop words (translated):\n")
  print(model_translated_terms[1:top_n_words, topic_translated])

  # Top documents
  theta_original <- model_original_theta[, topic_original]
  theta_translated <- model_translated_theta[, topic_translated]
  top_docs_original <- order(theta_original, decreasing = TRUE)[1:top_n_docs]
  top_docs_translated <- order(theta_translated, decreasing = TRUE)[1:top_n_docs]

  # Output
  cat("\nTop documents (original):\n")
  print(texts_original[top_docs_original])
  cat("\nTop documents (translated):\n")
  print(texts_translated[top_docs_translated])

}

# Apply function
topic_pair_number <- 6
top_n_words <- 20
top_n_docs <- 1
inspect_topic_pairs(results$model_original_terms,
                    results$model_translated_terms,
                    results$model_original_topicprobs,
                    results$model_translated_topicprobs,
                    results$unique_pairs, topic_pair_number,
                    translation_forth$original, translation_back$translated,
                    top_n_words, top_n_docs)
```

```
Comparing topic pair: Original = 6 , Translated = 1

Top words (original):
 [1] "kultur"                     "dekad"
 [3] "süd"                        "aktionsprogramm"
 [5] "partnerschaft"              "entwicklungsfinanzierung"
 [7] "entwicklung"                "habitat"
 [9] "naturkatastrophen"          "weltgipfel"
[11] "nachhaltig"                 "migrat"
[13] "weiterverfolgung"           "sport"
[15] "wenigsten"                  "vorbereitungsprozess"
[17] "entwickelten"               "agenda"
[19] "humanressourcen"            "menschenrechtserziehung"
```

```
Top words (translated):
 [1] "aufsichtsdienst"         "programmhaushalt"
 [3] "amtssprachen"            "handelsrecht"
 [5] "press"                   "konferenzausschuss"
 [7] "wiederaufgenommenen"     "koordinierungsausschuss"
 [9] "inspektionsgrupp"        "hauptabteilung"
[11] "intern"                  "zweijahresperiod"
[13] "institut"                "websit"
[15] "sekretariat"             "ausbildungsakademi"
[17] "manag"                   "qualität"
[19] "internen"                "missionen"

Top documents (original):
[1] "Siebenundfünfzigste Tagung Tagesordnungspunkt 97 Resolution der Generalversammlung [auf
Erklärung der Vereinten Nationen und die darin enthaltenen Entwicklungsziele, unter Begrüßur
Versammlung3, der Internationalen Konferenz über Entwicklungsfinanzierung, der Zweiten Weltv
12. März 1995 (auszugsweise Übersetzung des Dokuments A/CONF.166/9 vom 19. April 1995), Kap.
24/2, Anlage. Siehe Resolution 55/2. Abgedruckt in: Bericht der Zweiten Weltversammlung über
12. April 2002 (auszugsweise Übersetzung des Dokuments A/CONF.197/9), Kap. I, Resolution 1,
22. März 2002 (Veröffentlichung der Vereinten Nationen, Best.-Nr. E.02.II.A.7), Kap. I, Reso
4. September 2002 (auszugsweise Übersetzung des Dokuments A/CONF.199/20 vom 10. November 200

Top documents (translated):

Fitr und Id al-Adha berücksichtigt hat, und ersucht alle zwischenstaatlichen Organe, diese E
bei Bedarf“ Sitzungen abzuhalten, angemessene Konferenzdienste erhalten; 8. fordert die zwis
Hauptabteilung Generalversammlung und Konferenzmanagement zu halten, damit die während der E
Projekts, das die Integration der Informationstechnologie in die Sitzungsmanagement- und Dok
gleiche Rangstufe für gleiche Arbeit“ an den vier Hauptdienstorten befolgt wird; 3. bekräfti
sekretären zur Qualität der Konferenzdienste zu erkunden und der Generalversammlung über der
methoden und -verfahren der Konferenzdienste mit den einschlägigen Resolutionen der Generalv
Wochen-Regel und der Sechs-Wochen-Regel für die Herausgabe von Vorausdokumenten für Tagunger
Amt für interne Aufsichtsdienste, eine umfassende Überprüfung der bestehenden Sonderregelung
Sektion, und ersucht den Generalsekretär, mit Vorrang Abhilfe zu schaffen, unter anderem inc
second Session, Supplement No. A/62/161 und Corr.1 und 2 und Add.1 und Add.1/Corr.1. Officia
second Session, Supplement No. 32 (A/62/32), Anhang II. A/62/161 und Corr.1 und 2. Vereinte
"Zweiundsechzigste Sitzung Tagesordnungspunkt 131 Beschluss der Generalversammlung [basieren
Fitr und Eid, wie in den Resolutionen 53/208 A, 54/248, 55/222, 56/242, 57/283 B, 58/250, 59
Adha berücksichtigt und fordert alle zwischenstaatlichen Gremien auf, diese Entscheidungen z
Ziffer 38 des Berichts des Generalsekretärs zur Kenntnis und fordert ihn auf, dafür zu sorge
nach Bedarf“ abzuhalten, angemessene Konferenzdienste erhalten; 8. fordert die zwischenstaat
Umbauplans zu berücksichtigen; 2. ersucht den Generalsekretär, sicherzustellen, dass die Arb
G Generell muss eine angemessene informationstechnische Unterstützung der Dokumentationsdier
Wiederherstellungsplans ununterbrochen arbeiten können. 7. stellt fest, dass ein Teil des Ko
Ressourcen der Hauptabteilung Generalversammlung und Konferenzmanagement während der Umsetzu
Wiederherstellungsplans vorübergehend in alternativen Einrichtungen untergebracht werden, ur
Einrichtungen der Abteilung weiterhin aufrechtzuerhalten, die globale Informationstechnologi
Initiative umzusetzen und hochwertige Konferenzdienste bereitzustellen; Integriertes globale
Lösungen des Generals A Montage zu erledigen; 4. ersucht den Generalsekretär, sicherzustelle
sekretären zur Qualität der Konferenzdienste zu sammeln und zu analysieren und der Generalve
Wochen-Regel und der Sechs-Wochen-Regel für die Herausgabe von Unterlagen für Vorabsitzunger
```

```
dritte Sitzung durch den Konferenzausschuss; 5. bringt seine anhaltende Besorgnis über die h
steuerung aus einer Gesamtsystemperspektive vorgeschlagen hat, und sieht der Vorlage der Inc
```

Based on the output, we observe that the matched topics in pair number 6 use different top words. However, it is unclear whether these differences arise from translation errors or from artifacts of the topic modeling process itself (Maier et al. 2022).

To investigate this, we extract the context of selected words from both the original and translated data using the KWIC (Key Word in Context) method. This allows us to examine the frequency of each word and inspect its surrounding text, helping to identify potential translation issues. As an example, we focus on the word "Menschenrechtserziehung" (*human rights education*).

```r
# KWIC function
get_kwic <- function(texts, word, window = 5) {
  word_pattern <- paste0("\\b", word, "\\b")

  unlist(lapply(texts, function(txt) {
    if (is.na(txt) || txt == "") return(NA_character_)

    words <- unlist(str_split(txt, "\\s+"))
    positions <- which(str_detect(words, regex(word_pattern, ignore_case = TRUE)))

    if(length(positions) == 0) return(NA_character_)

    contexts <- sapply(positions, function(pos) {
      start <- max(1, pos - window)
      end <- min(length(words), pos + window)
      paste(words[start:end], collapse = " ")
    })

    paste(contexts, collapse = " | ")
  }))
}

# Main function
extract_kwic <- function(word_of_interest, translation_back, translation_forth, window = 5,

  # Occurrences in "original"
  forth_kwic <- translation_forth %>%
    filter(str_detect(.data[["original"]], regex(word_of_interest, ignore_case = TRUE))) %>%
    select(all_of(id_col), original, translated) %>%
    mutate(
      original_kwic = get_kwic(original, word_of_interest, window)
    )

  # Occurrences in "translated"
  back_kwic <- translation_back %>%
    filter(str_detect(.data[["translated"]], regex(word_of_interest, ignore_case = TRUE))) %
    select(all_of(id_col), original, translated) %>%
    mutate(
      translated_kwic = get_kwic(translated, word_of_interest, window)
    )
```

```r
  # Combine into one data frame
  kwic_df <- full_join(forth_kwic, back_kwic, by = "id")
  kwic_df <- kwic_df %>%
    select(all_of(id_col), original_kwic, translated_kwic)

  return(kwic_df)
}

# Example usage
word_of_interest <- "menschenrechtserziehung"
result_kwic <- extract_kwic(word_of_interest, translation_back, translation_forth, window =

print(result_kwic)
```

```
# A tibble: 16 × 3
      id original_kwic                                                    translated_kwic
   <int> <chr>                                                            <chr>
 1   137 Gewährung von Schutz, Menschenrechtserziehung oder Ver… Forschung, Fak
…
 2   158 dem Gebiet der Menschenrechtserziehung eingeleitet. Ei… Vereinten Nati
…
 3   320 Dritten Ausschusses (A/56/583/Add.2)] Menschenrechtser… Dritten Aussch
…
 4   335 angelegte Strategie für Menschenrechtserziehung umfass… umfassende Str
…
 5   342 Organisationen auf, die Menschenrechtserziehung sowie … Nichtregierung
…
 6   389 Strategien für die Menschenrechtserziehung auszuarbeit… wirksame Strat
…
 7   405 Vereinten Nationen für Menschenrechtserziehung (1995-2… Vereinten Nati
…
 8   564 Dritten Ausschusses (A/57/556/Add.2)] Menschenrechtser… Dritten Aussch
…
 9   746 Vereinten Nationen für Menschenrechtserziehung 1995-20… Add.1)] Weltpr
…
10   753 wie wichtig die Menschenrechtserziehung und -ausbildun… die Bedeutung
…
11   809 Organisationen auf, die Menschenrechtserziehung sowie … zur Förderung
…
12   813 Vereinten Nationen für Menschenrechtserziehung (1995-2… Vereinten Nati
…
13   881 anderem bei der Menschenrechtserziehung und -ausbildun… auch bei der M
…
14   898 Vereinten Nationen für Menschenrechtserziehung 1995-20… Vereinten Nati
…
15   923 Vereinten Nationen für Menschenrechtserziehung (1995-2… Vereinten Nati
…
16   933 Vereinten Nationen für Menschenrechtserziehung 1995-20… <NA>
```

We observe that the word is also present in the translated dataset, remaining unchanged after the back-and-forth translation in 16 out of 17 cases. Furthermore, the meanings revealed by the KWIC analysis remain consistent (interpreted based on German content), indicating that no severe translation issues are present regarding this particular top word. Further words can be examined using the same approach to further validate this observation.

4. Conclusion and recommendations

This tutorial provides a foundation for evaluating the quality of machine translation in the context of multilingual topic modeling. The proposed metrics enable a comprehensive evaluation of impacts of machine translation on topic modeling outcomes and can be applied to other study cases when examining back-and-forth translation outputs. The metrics applied in the study, adapted from De Vries, Schoonvelde, and Schumacher (2018), are valuable for input validation and offer a basis for future evaluations, contributing to the growing body of literature and emerging guidelines on multilingual analysis, while acknowledging critiques regarding the privileging of English over other languages in research (Lind et al. 2022; Baden et al. 2022).

A key limitation of this tutorial is that it considers only one language pair and relies solely on the UN text corpus. While there are studies using other types of data and languages (e.g. Maier et al. 2022), further cases need to be explored, possibly enriched through closer involvement of language and cultural expertise. Moreover, the abilities of advanced approaches, including embedding-based methods (e.g. Grootendorst 2022), for multilingual topic modeling remain uncovered and should be addressed in future studied. Nevertheless, the application of this tool remains relevant in contexts of probabilistic approaches, which remain widely used, interpretable, and allow control over data quality at each stage of the analysis.

References

DOI

https://doi.org/10.71627/topling

Baden, Christian, Alona Dolinsky, Fabienne Lind, Christian Pipal, Martijn Schoonvelde, Guy Shababo, and Mariken A. C. G. Van der Velden. 2022. "Integrated Standards and Context-Sensitive Recommendations for the Validation of Multilingual Computational Text Analysis." *Technical Report Deliverable 6.2.*

Baden, Christian, and K Stalpouskaya. 2015. "Common Methodological Framework: Content Analysis. A Mixed-Methods Strategy for Comparatively, Diachronically Analyzing Conflict Discourse." *INFOCORE Working Paper*. https://www.infocore.eu/wp-content/uploads/2016/02/Methodological-Paper-MWG-CA_final.pdf.

Benoit, Kenneth, and Akitaka Matsuo. 2017. "Spacyr: Wrapper to the 'Spacy' 'NLP' Library (1.2.1)." https://CRAN.R-project.org/package=spacyr.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. "Quanteda: An R Package for the Quantitative Analysis of Textual Data." *Journal of Open Source Software* 3 (30): 774. https://doi.org/10.21105/joss.00774.

Bernhard, Jana, Martin Teuffenbach, and Hajo G. Boomgaarden. 2023. "Topic Model Validation Methods and Their Impact on Model Selection and Evaluation." *Computational Communication Research* 5 (1): 1. https://doi.org/10.5117/CCR2023.1.13.BERN.

Chan, Chung-Hong, Jing Zeng, Hartmut Wessler, Marc Jungblut, Kasper Welbers, Joseph W Bajjalieh, Wouter Van Atteveldt, and Scott L. Althaus. 2020. "Reproducible Extraction of Cross-Lingual Topics (Rectr)." *Communication Methods and Measures* 14 (4): 285–305. https://doi.org/10.1080/19312458.2020.1812555.

De Vries, Erik, Martijn Schoonvelde, and Gijs Schumacher. 2018. "No Longer Lost in Translation: Evidence That Google Translate Works for Comparative Bag-of-Words Text Applications." *Political Analysis* 26 (4): 417–30. https://doi.org/10.1017/pan.2018.26.

Eisele, Andreas, and Yu Chen. 2010. "MultiUN: A Multilingual Corpus from United Nation Documents." *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, 2868–72.

Grootendorst, Maarten. 2022. "BERTopic: Neural Topic Modeling with a Class-Based TF-IDF Procedure." arXiv. https://doi.org/10.48550/ARXIV.2203.05794.

Jacobi, Carina, Wouter Van Atteveldt, and Kasper Welbers. 2016. "Quantitative Analysis of Large Amounts of Journalistic Texts Using Topic Modelling." *Digital Journalism* 4 (1): 89–106. https://doi.org/10.1080/21670811.2015.1093271.

Lind, Fabienne, Jakob-Moritz Eberl, Olga Eisele, Tobias Heidenreich, Sebastian Galyga, and Hajo G. Boomgaarden. 2022. "Building the Bridge: Topic Modeling for Comparative Research." *Communication Methods and Measures* 16 (2): 96–114. https://doi.org/10.1080/19312458.2021.1965973.

Maier, Daniel, Christian Baden, Daniela Stoltenberg, Maya De Vries-Kedem, and Annie Waldherr. 2022. "Machine Translation Vs. Multilingual Dictionaries Assessing Two Strategies for the Topic Modeling of Multilingual Text Collections." *Communication Methods and Measures* 16 (1): 19–38. https://doi.org/10.1080/19312458.2021.1955845.

Maier, Daniel, A. Waldherr, P. Miltner, G. Wiedemann, A. Niekler, A. Keinert, B. Pfetsch, et al. 2018. "Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology." *Communication Methods and Measures* 12 (2-3): 93–118. https://doi.org/10.1080/19312458.2018.1430754.

Peters, Yannik, and Kunjan Shah. 2024. "Comparing Tools and Workflows for Data Quality in Text Preprocessing." Methods Hub. https://doi.org/10.71627/TEXTPREP.

Reber, Ueli. 2019. "Overcoming Language Barriers: Assessing the Potential of Machine Translation and Topic Modeling for the Comparative Analysis of Multilingual Text Corpora." *Communication Methods and Measures* 13 (2): 102–25. https://doi.org/10.1080/19312458.2018.1555798.

Roberts, Margaret E., Brandon M. Stewart, and Dustin Tingley. 2019. "**Stm** : An *r* Package for Structural Topic Models." *Journal of Statistical Software* 91 (2). https://doi.org/10.18637/jss.v091.i02.

Van Atteveldt, Wouter, Damian Trilling, and Carlos Arcila Calderón. 2022. *Computational Analysis of Communication*. Wiley Blackwell.