Test

```r
rm(list=ls(all=T))
# uncomment this if running first time on local
# To rum on AWS, always uncomment this
# install.packages(c( "tidyverse", "readxl", "tm", "topicmodels", "NLP"))
```

```r
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.6.3
```

```
## Loading required package: NLP
```

```
## Warning: package 'NLP' was built under R version 3.6.3
```

```r
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 3.6.3
```

```r
library(NLP)
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.6.3
```

```r
# options(stringsAsFactors = F)
```

Loading statements

```r
docs <- readxl::read_excel("E:/Projects/modern_slavery_registry/data/sheets/subset_data.xlsx")
docs <- docs[["final_statement_cleaned"]]
# https://stackoverflow.com/questions/47406555/error-faced-while-using-tm-packages-vcorpus-in-r
docs <- data.frame(doc_id=c(1:length(docs)), text=docs)
```

Preparing document-term matrix for n-grams [1-n]

```
# https://stackoverflow.com/questions/45697840/custom-tokenizer-in-tm-package-r-not-working
corpus <- VCorpus(DataframeSource(docs))

# https://stackoverflow.com/questions/52207021/r-how-to-apply-terms-from-training-document-term-
matrix-dtm-to-test-dtm-bot
custom_tokenizer <- function(x) unlist(lapply(NLP::ngrams(words(x), 1:2), paste, collapse = " "
), use.names = FALSE)


MIN_DF <- 10
MAX_DF <- 1000
dtm <- DocumentTermMatrix(
  corpus,
  control = list(
    tokenize=custom_tokenizer,
    bounds = list(global = c(MIN_DF, MAX_DF))))
```

Document-term dimension

```
cat("docs:", dim(dtm)[1], "vocab-size:", dim(dtm)[2])
```

```
## docs: 9993 vocab-size: 47987
```

Pre-process document-term matrix

```
# due to vocabulary pruning, we have empty rows in our DTM
# LDA does not like this. So we remove those docs from the
# DTM and the metadata
sel_idx <- slam::row_sums(dtm) > 0
dtm <- dtm[sel_idx, ]
```

Running LDA model

```
# number of topics
NUM_TOPICS <- 10
# set random number generator seed
set.seed(40)
NUM_ITER = 500
# compute the LDA model, inference via 500 iterations of Gibbs sampling
topicModel <- LDA(
  x=dtm,
  k=NUM_TOPICS,
  method="Gibbs",
  control=list(iter = NUM_ITER, verbose = 25))
```

```
## K = 10; V = 47987; M = 9993
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

Getting top n words for each topic

```
NUM_TOP_WORDS = 10
terms(topicModel, NUM_TOP_WORDS)
```

```
##       Topic 1            Topic 2           Topic 3     Topic 4
##  [1,] "2015 act"         "2015 set"        "germany"   "potential risk"
##  [2,] "procure"          "traffic take"    "france"    "reputable"
##  [3,] "plc"              "within business" "ireland"   "circumstance"
##  [4,] "asset"            "place within"    "australia" "statement set"
##  [5,] "outsource"        "eligibility"     "usa"       "vehicle"
##  [6,] "clause"           "commit act"      "africa"    "commit prevent"
##  [7,] "regulate"         "continue take"   "canada"    "put place"
##  [8,] "wholly"           "minimum wage"    "japan"     "retaliation"
##  [9,] "executive officer" "reprisal"       "spain"     "easy"
## [10,] "bank"             "31st"            "italy"     "aim ensure"
##       Topic 5             Topic 6      Topic 7      Topic 8
##  [1,] "identity"          "award"      "store"      "target"
##  [2,] "physical"          "investor"   "shop"       "strengthen"
##  [3,] "threat"            "charity"    "return"     "compact"
##  [4,] "sedex"             "research"   "accessory"  "embed"
##  [5,] "recruit"           "school"     "send"       "align"
##  [6,] "knowingly"         "innovation" "click"      "launch"
##  [7,] "sexual"            "vision"     "notice"     "channel"
##  [8,] "permanent"         "study"      "gift"       "goal"
##  [9,] "equal opportunity"  "tender"    "collection" "collaboration"
## [10,] "accordance section" "energy"    "card"       "map"
##       Topic 9                 Topic 10
##  [1,] "whistle blower"        "chain act"
##  [2,] "control ensure"        "transparency supply"
##  [3,] "high level"            "california transparency"
##  [4,] "level understand"      "disclose"
##  [5,] "initiative identify"   "involuntary"
##  [6,] "exploit"               "among"
##  [7,] "organization structure" "inc"
##  [8,] "understand risk"       "corrective"
##  [9,] "anti policy"           "comply applicable"
## [10,] "policy reflect"        "prison"
```

```
# have a look a some of the results (posterior distributions)
tmResult <- posterior(topicModel)
# format of the resulting object
attributes(tmResult)
```

```
## $names
## [1] "terms"  "topics"
```

```
# topics are probability distribtions over the entire vocabulary
beta <- tmResult$terms   # get beta from results
dim(beta)                # K distributions over nTerms(DTM) terms
```

```
## [1]    10 47987
```

```
# aggregated beta distribution
rowSums(beta)
```

```
##  1  2  3  4  5  6  7  8  9 10
##  1  1  1  1  1  1  1  1  1  1
```

Computing perplexity for trained model

```
perplexity(topicModel, dtm)
```

```
## K = 10; V = 47987; M = 9993
## Sampling 500 iterations!
## Iteration 25 ...
## Iteration 50 ...
## Iteration 75 ...
## Iteration 100 ...
## Iteration 125 ...
## Iteration 150 ...
## Iteration 175 ...
## Iteration 200 ...
## Iteration 225 ...
## Iteration 250 ...
## Iteration 275 ...
## Iteration 300 ...
## Iteration 325 ...
## Iteration 350 ...
## Iteration 375 ...
## Iteration 400 ...
## Iteration 425 ...
## Iteration 450 ...
## Iteration 475 ...
## Iteration 500 ...
## Gibbs sampling completed!
```

```
## [1] 13157
```