Acled Technical Assignment - Human Rights Text Analysis

Load Libraries

```
library(knitr)
library(tm)
```

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

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::annotate() masks NLP::annotate()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(quanteda)
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "pcorMatrix" of class "replValueSp"; definition not updated
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "pcorMatrix" of class "xMatrix"; definition not updated
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "pcorMatrix" of class "mMatrix"; definition not updated
```

```
## Package version: 3.3.1
## Unicode version: 14.0
## ICU version: 71.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
##
## Attaching package: 'quanteda'
##
## The following object is masked from 'package:tm':
##
##     stopwords
##
## The following objects are masked from 'package:NLP':
##
##     meta, meta<-
```

```
library(readtext)
```

```
##
## Attaching package: 'readtext'
##
## The following object is masked from 'package:quanteda':
##
##     texts
```

```
library(stm)
```

```
## stm v1.3.6 successfully loaded. See ?stm for help.
##  Papers, resources, and other materials at structuraltopicmodel.com
```

```
library(tidytext)
library(ggthemes)
library(quanteda.textplots)
```

Read Data

Read data – a folder which contains a txt file for each report labelled in the format reportname.txt.

For this analysis I used a dataset from Christopher et.al published on the Harvard Dataverse. Here is the citation for the same.:

Christopher J. Fariss; Fridolin J. Linder; Zachary M. Jones; Charles D. Crabtree; Megan A. Biek; Ana-Sophia M. Ross; Taranamol Kaur; Michael Tsai, 2015, "Human Rights Texts: Converting Human Rights Primary Source Documents into Data", https://doi.org/10.7910/DVN/IAH8OY, Harvard Dataverse, V3

```
#Read Data
text <- readtext("dataverse_files_acled/dataverse/*.txt")
```

Sampling

Due to computational limitations, I take a simple random sample of 1000 texts.

```
#Sample
articles <- text %>%
  sample_n(1000)
```

Pre Processing

This data was analyzed using Quanteda and tidyverse in R. After sampling the number of documents, I began pre-processing the data. I create tokens to reduce the text into smaller, more interpretable objects. Thereafter, I perform a series of common pre-processing practices that reduce noise, increase computational efficiency, and make topic models generate topics that are concise and coherent. Some of these pre-processing steps include removing punctuation, numbers, urls and stop words common in the English language. I also create compound tokens to indicate the combination of words being in unison such as – "human rights" "u.s" etc. After performing a series of pre-processing texts, I create a document frequency matrix that describes frequency of terms in each document.

Given the limitations of processing-power, please note pre-processing is an iterative step and this would become clearer when the topics are generated in the end. Despite performing these pre-processing steps, the data often carries noise because of differing writing styles, context of the themes and topics being analyzed etc. To ideate on pre-processing, I would like to discuss this further with any technical stakeholders and substantive experts who are well versed with literature in this field of research.

```
#select text column from articles
tokens <- articles$text %>%
  #tokenize to words
          tokens(what = "word",
                 #remove punctuation
                 remove_punct = TRUE,
                 #remove numbers
                 remove_numbers = TRUE,
                 #remove urls
                 remove_url = TRUE
                 ) %>%
  #change all tokens to lowercase
  tokens_tolower() %>%
  #remove common stop words from the english language
  tokens_remove(stopwords("english")) %>%
  #stem using quanteda's language stemmer
  #lemmetization potential here#
  tokens_wordstem(language = quanteda_options("language_stemmer")) %>%
  #compound token to keep the word "human right" together
  #add un here
  tokens_compound(pattern = c("human right*", "u.s.*", "domestic violence*", "un*"))

#applying relative pruning, create document feature matrix where the minimum term frequency is set to 3
dfm <- dfm_trim(dfm(tokens), min_docfreq = 0.30, max_docfreq = 0.90, min_termfreq = 75, docfreq_type =
```

```
## Removing features occurring:
```

```
##   - fewer than 75 times: 76,711
```

```
##    - in fewer than 300 documents: 79,687

##    - in more than 900 documents: 7

##    Total features removed: 79,694 (99.1%).
```

```
#remove additional characters
dfm <- dfm_remove(dfm,c("<",">", "however", "although", "$", "also"))
```

```
textplot_wordcloud(dfm, max_words = 50, random_order = TRUE, color = "#0086b3")
```


```
#convert dfm into a stm structure that is compatible with analysis in library(stm)
dfm_stm <- convert(dfm, to = "stm")
```

Modelling

To run a Structured Topic Model, I begin by running a search K function which enables me to test the optimal number of topics that can be generated from this text. These Ks are usually analyzed using evaluation metrics for goodness of fit like Coherence, Residuals, Lower Bound and Exclusivity. K = 7 seems to be an optimal fit for the model from an initial look at the evaluation metrics however, this is also something I would usually discuss with stakeholders or fellow technical members of the team. After running the model for K = 7, I plot the proportion of topical prevalence in the texts and the top 7 words that exist in each topic. Lastly, I also create a gamma matrix which gives me the probability of each document being associated with a topic.

```r
#Select the number of K to search optimal number of topics
K = c(5,6,7,8,9,10)

#Run Search K model to check goodness of fit for each K
model_test <- searchK(dfm_stm$documents, dfm_stm$vocab, K = K, verbose = TRUE)
```

```r
# Plot Eval Metrics for checking model fit
plot <- data.frame("K" = K,
                   "Coherence" = unlist(model_test$results$semcoh),
                   "Exclusivity" = unlist(model_test$results$exclus),
                   "Residual" =  unlist(model_test$results$residual),
                   "Lower Bound" = unlist(model_test$results$lbound))

# Reshape to long format
library("reshape2")
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

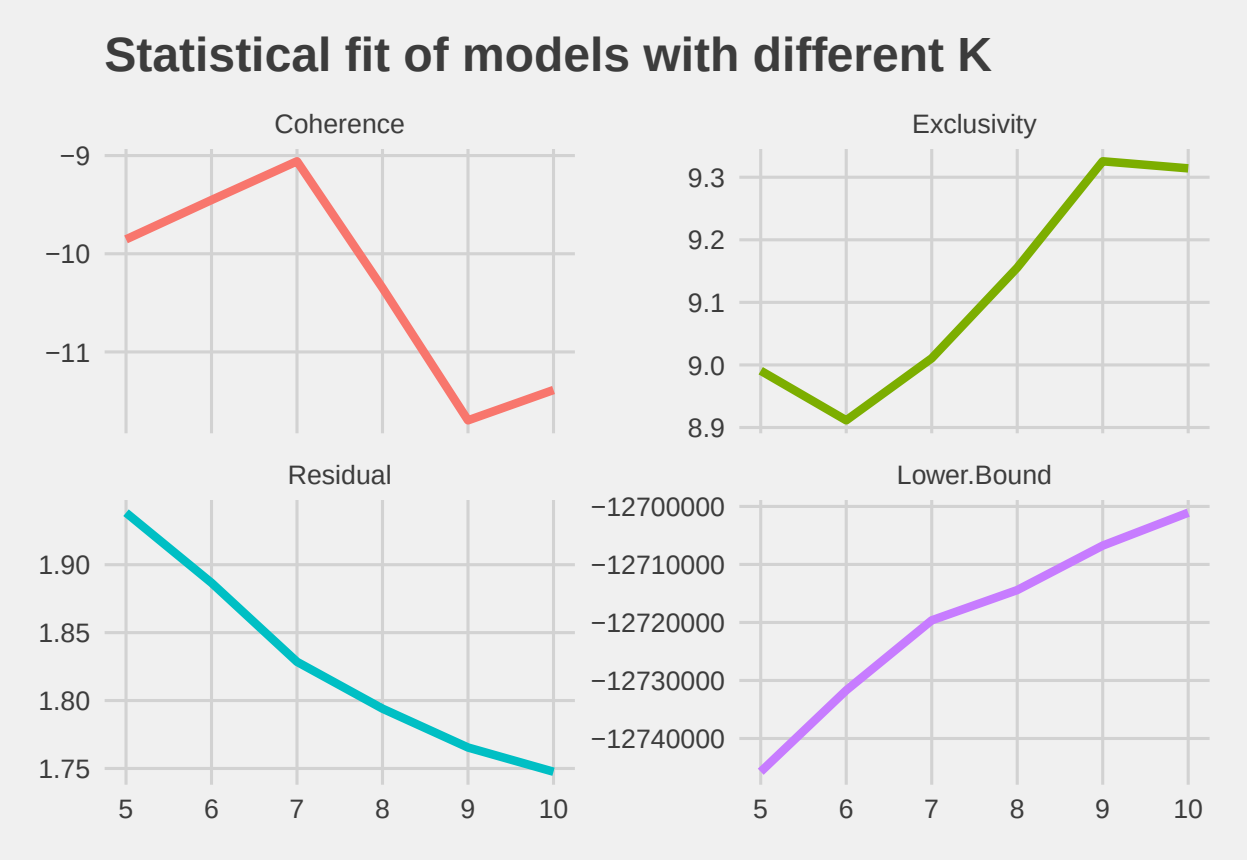
```r
plot <- melt(plot, id=c("K"))
plot
```

```
##     K    variable         value
## 1   5   Coherence -9.853421e+00
## 2   6   Coherence -9.452427e+00
## 3   7   Coherence -9.058166e+00
## 4   8   Coherence -1.034842e+01
## 5   9   Coherence -1.169689e+01
## 6  10   Coherence -1.138787e+01
## 7   5 Exclusivity  8.990733e+00
## 8   6 Exclusivity  8.911444e+00
## 9   7 Exclusivity  9.010657e+00
## 10  8 Exclusivity  9.155367e+00
## 11  9 Exclusivity  9.325470e+00
## 12 10 Exclusivity  9.314189e+00
## 13  5    Residual  1.938270e+00
## 14  6    Residual  1.886643e+00
## 15  7    Residual  1.828504e+00
## 16  8    Residual  1.794040e+00
## 17  9    Residual  1.765425e+00
## 18 10    Residual  1.747585e+00
## 19  5 Lower.Bound -1.274573e+07
## 20  6 Lower.Bound -1.273170e+07
## 21  7 Lower.Bound -1.271963e+07
## 22  8 Lower.Bound -1.271442e+07
## 23  9 Lower.Bound -1.270675e+07
## 24 10 Lower.Bound -1.270108e+07
```

```
library("ggplot2")
fit_stats <- ggplot(plot, aes(K, value, color = variable)) +
  geom_line(size = 1.5, show.legend = FALSE) +
  facet_wrap(~variable,scales = "free_y") +
  labs(x = "Number of topics K",
       title = "Statistical fit of models with different K")+
  theme_fivethirtyeight()
```

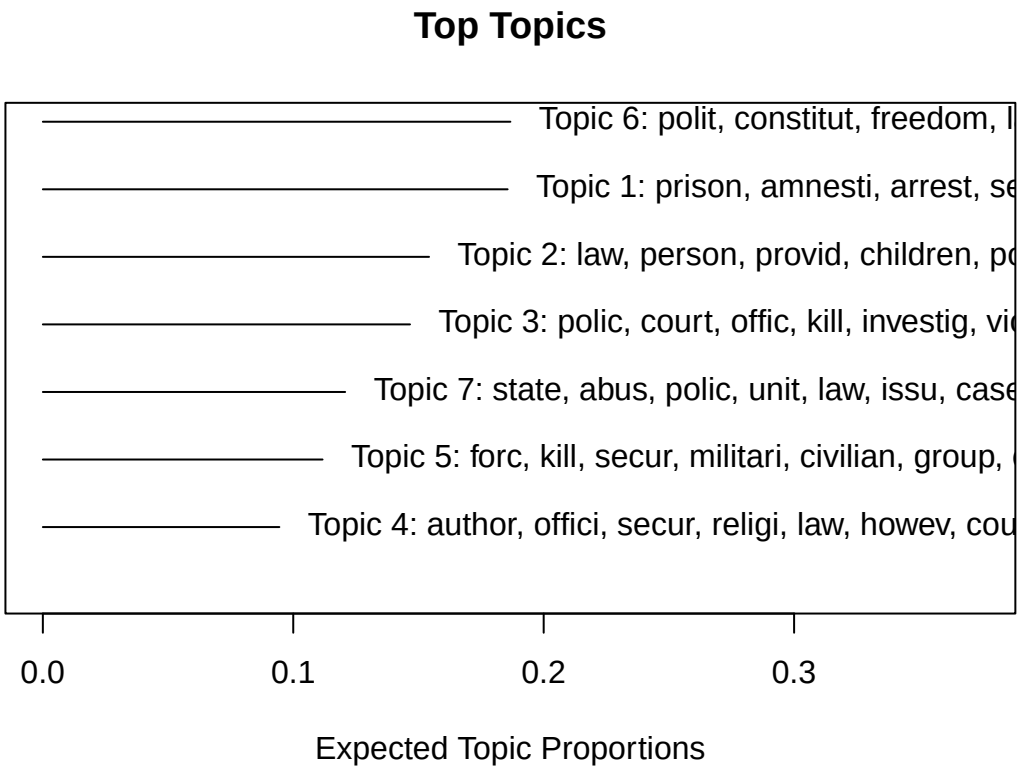
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
print(fit_stats)
```




```
# Run the model with K = best fit scores
model <- stm(documents = dfm_stm$documents,
             vocab = dfm_stm$vocab,
             K = 7,
             verbose = TRUE)
```

```r
#plot the topics and the top 7 words from the topic
plot.STM(model, "summary", n=7)
```




```r
# For each topic, print the first seven common words, use FREX score to evaluate model
print(labelTopics(model,topics = c(1:7), n=7))
```

```
## Topic 1 Top Words:
##       Highest Prob: prison, amnesti, arrest, sentenc, releas, peopl, death
##       FREX: amnesti, prison, releas, sentenc, peopl, execut, imprison
##       Lift: amnesti, imprison, appar, execut, inquiri, beaten, said
##       Score: amnesti, prison, peopl, sentenc, death, releas, arrest
## Topic 2 Top Words:
##       Highest Prob: law, person, provid, children, polic, labor, case
##       FREX: children, provid, prohibit, traffick, child, person, labor
##       Lift: workweek, disabl, traffick, child, prohibit, children, sexual
##       Score: workweek, labor, traffick, percent, disabl, child, sexual
## Topic 3 Top Words:
##       Highest Prob: polic, court, offic, kill, investig, violenc, tortur
##       FREX: polic, crime, violenc, offic, un, shot, investig
##       Lift: shot, un, risk, perpetr, conclud, crime, fail
##       Score: shot, un, polic, violenc, sexual, kill, peopl
## Topic 4 Top Words:
##       Highest Prob: author, offici, secur, religi, law, howev, court
##       FREX: religi, foreign, restrict, author, see, offici, newspap
##       Lift: whose, religi, permiss, church, materi, regist, newspap
```

```
##       Score: whose, religi, see, activist, section, author, regist
## Topic 5 Top Words:
##       Highest Prob: forc, kill, secur, militari, civilian, group, continu
##       FREX: civilian, war, arm, kill, forc, refuge, soldier
##       Lift: war, soldier, humanitarian, civilian, conflict, camp, armi
##       Score: war, soldier, arm, kill, civilian, forc, attack
## Topic 6 Top Words:
##       Highest Prob: polit, constitut, freedom, labor, union, public, parti
##       FREX: respect, constitut, guarante, c, religion, b, freedom
##       Lift: guarante, emigr, economi, b, e, f, freeli
##       Score: guarante, labor, percent, b, wage, c, emigr
## Topic 7 Top Words:
##       Highest Prob: state, abus, polic, unit, law, issu, case
##       FREX: unit, administr, monitor, depart, develop, note, issu
##       Lift: note, toward, import, world, attent, unit, administr
##       Score: note, state, monitor, unit, abus, labor, effort
```

```r
#Save top 20 features across topics and forms of weighting
labels <- labelTopics(model, n=30)
#only keep FREX weighting
topwords <- data.frame("features" = t(labels$frex))
#assign topic number as column name
colnames(topwords) <- paste("Topics", c(1:7))
#Return the result
print(topwords[1:7])
```

```
##    Topics 1  Topics 2  Topics 3   Topics 4     Topics 5  Topics 6  Topics 7
## 1   amnesti  children     polic     religi     civilian   respect      unit
## 2    prison    provid     crime    foreign          war constitut administr
## 3    releas  prohibit   violenc   restrict          arm  guarante   monitor
## 4   sentenc  traffick     offic     author         kill         c    depart
## 5     peopl     child        un        see         forc  religion   develop
## 6    execut    person      shot     offici       refuge         b      note
## 7  imprison     labor  investig    newspap      soldier   freedom      issu
## 8     trial    employ    crimin       deni       attack      free    polici
## 9     death    disabl    justic     church     militari      wage    effort
## 10     held discrimin      kill journalist         area   percent     state
## 11   arrest   percent    victim     permit         armi     union      role
## 12  detaine       age      fail      local     conflict    privat    import
## 13   detain   problem    attack     accord humanitarian     labor     world
## 14   appeal  ministri      face     regist         camp         d      abus
## 15   tortur  approxim     alleg      whose        secur    employ      even
## 16     said   general     court      howev        group    econom       aid
## 17    charg       law    tortur   demonstr       region     grant    violat
## 18  convict    domest   protest    opposit       ethnic   practic   million
## 19  custodi    sexual     death   activist       villag      must     first
## 20  without    enforc    sexual      allow         oper    tradit   commiss
## 21   detent   practic      head     requir     thousand       can    reform
## 22       end    worker   commiss    permiss      opposit    travel    toward
## 23    alleg   corrupt  prosecut      media      support     emigr   concern
## 24    least    school   penalti      resid         mani    within  document
## 25    appar   minimum    threat    citizen       return   assembl     press
## 26    three     women     feder      refus   throughout    social    improv
## 27   decemb   section communiti      secur        sever    system       new
```

```
## 28    other    sector    murder     obtain       border    sector      well
## 29      two    access   protect      claim      countri   minimum      step
## 30      die      educ    defend    section   journalist   economi    assist
```

```
#probability of each document being associated with each topic (Sample head(10))
theta <- make.dt(model)
theta[1:10,1:8]
```

```
##     docnum      Topic1       Topic2      Topic3      Topic4     Topic5
##  1:      1 0.161884571 0.0042926785 0.557603905 0.038177911 0.09583516
##  2:      2 0.048075417 0.1250618883 0.032969797 0.024797983 0.27841687
##  3:      3 0.007431762 0.0076406366 0.038671579 0.009023799 0.19265399
##  4:      4 0.058398862 0.0007046022 0.066992387 0.065057561 0.27103448
##  5:      5 0.037683251 0.0112496390 0.014518412 0.022267327 0.06370809
##  6:      6 0.082783607 0.0160215113 0.692751781 0.021406037 0.01678153
##  7:      7 0.005182059 0.4835171716 0.003773237 0.123492566 0.34911135
##  8:      8 0.462951532 0.0003982533 0.165517541 0.008333110 0.35615877
##  9:      9 0.014823575 0.5290947437 0.105572456 0.113150680 0.10630063
## 10:     10 0.022689057 0.1138448750 0.006568352 0.014250901 0.00732270
##          Topic6      Topic7
##  1: 0.012676316 0.129529456
##  2: 0.401849444 0.088828601
##  3: 0.038221916 0.706356315
##  4: 0.035242645 0.502569459
##  5: 0.743661554 0.106911723
##  6: 0.042754249 0.127501286
##  7: 0.031250906 0.003672715
##  8: 0.001186934 0.005453856
##  9: 0.028887928 0.102169989
## 10: 0.814408356 0.020915759
```

Visualization and Insights (Part 2 of the Technical Assignment)

In this section, I develop some exploratory graphs to see the topic prevalance in our sample. Fig 1.1 shows the proportion of topical prevalance in our sample, and I overlay the top 7 words the occur in each topic. Fig 1.2 goes a step further, and gives the proportion of the words occurring in each topic. These two graphs serve as an inital exploration point to observe keywords and see if there are thematic trends prevelant in the data. In addition, I also create a time-series which show the number of texts published by each organization in each year. Here n = 14190, which is the entirety of the dataset.

```
#convert model into tidy tibble
model_beta <- tidy(model)
head(model_beta)
```

```
## # A tibble: 6 x 3
##   topic term       beta
##   <int> <chr>     <dbl>
## 1     1 abl   0.000107
## 2     2 abl   0.000207
## 3     3 abl   0.0000121
## 4     4 abl   0.000705
## 5     5 abl   0.000484
## 6     6 abl   0.000717
```

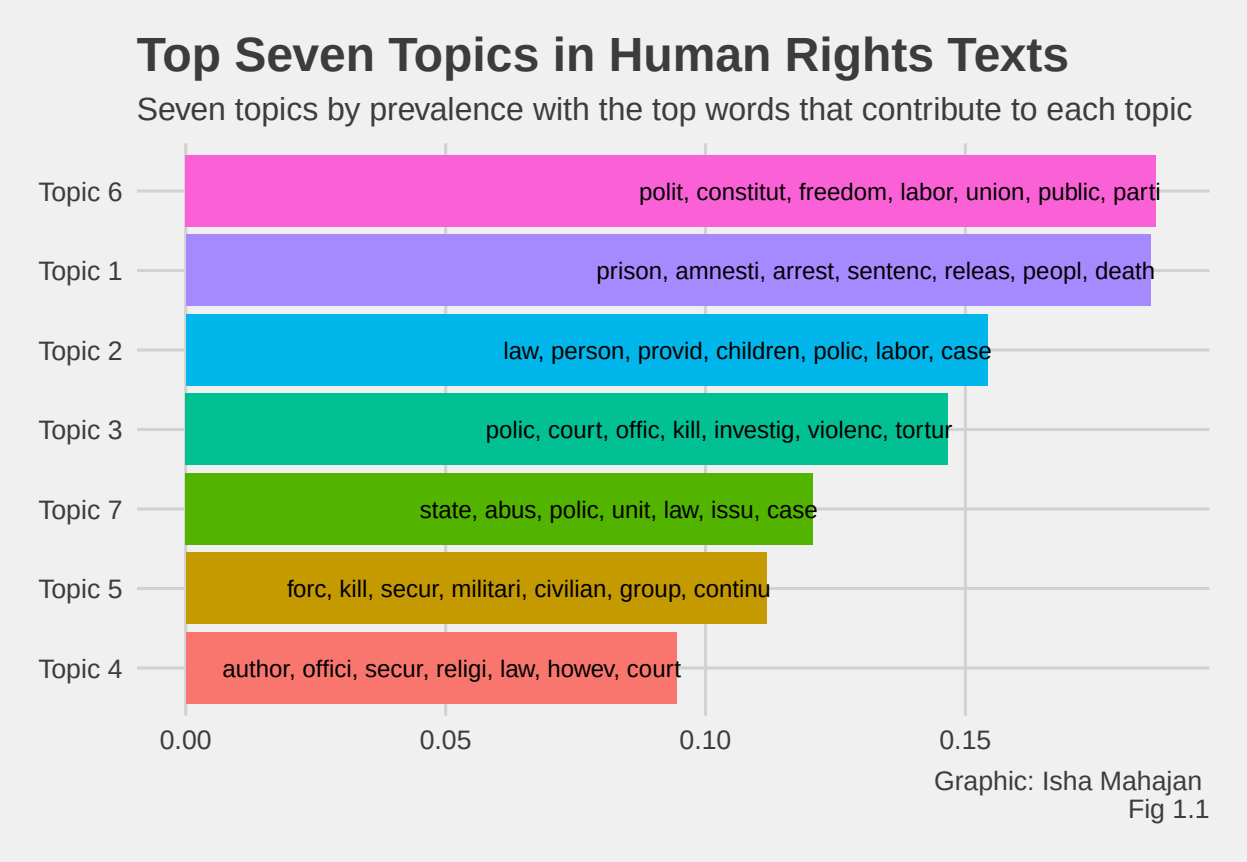
```r
#covert model into tibble gamma matrix -- probability of each document being associated with a topic
model_gamma <- tidy(model, matrix = "gamma",
                    document_names = rownames(articles))
model_gamma
```

```
## # A tibble: 7,000 x 3
##    document topic   gamma
##    <chr>    <int>   <dbl>
##  1 1            1 0.162
##  2 2            1 0.0481
##  3 3            1 0.00743
##  4 4            1 0.0584
##  5 5            1 0.0377
##  6 6            1 0.0828
##  7 7            1 0.00518
##  8 8            1 0.463
##  9 9            1 0.0148
## 10 10           1 0.0227
## # i 6,990 more rows
```

```r
top_terms <- model_beta%>%
  arrange(beta) %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  arrange(-beta) %>%
  select(topic, term) %>%
  summarise(terms = list(term)) %>%
  mutate(terms = map(terms, paste, collapse = ", ")) %>%
  unnest(cols = c(terms))

gamma_terms <- model_gamma %>%
  group_by(topic) %>%
  summarise(gamma = mean(gamma)) %>%
  arrange(desc(gamma)) %>%
  left_join(top_terms, by = "topic") %>%
  mutate(topic = paste0("Topic ", topic),
         topic = reorder(topic, gamma))

figone_one <- gamma_terms %>%
  top_n(8, gamma) %>%
  ggplot(aes(topic, gamma, label = terms, fill = topic)) +
  geom_col(show.legend = FALSE) +
  geom_text(hjust = 1, nudge_y = 0.0009, size = 3) +
  coord_flip() +
  theme_hc() +
  theme(plot.title = element_text(size = 12)) +
  labs(x = NULL, y = expression(gamma),
       title = "Top Seven Topics in Human Rights Texts",
       subtitle = "Seven topics by prevalence with the top words that contribute to each topic",
       caption = "Graphic: Isha Mahajan \nFig 1.1")+
  theme_fivethirtyeight()
print(figone_one)
```




```
#ggsave(figone_one, "fig1.1.png", dpi = 400)
```

```
figeone_two <- td_beta <- tidytext::tidy(model)
td_beta %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
 ungroup() %>%
    mutate(topic = paste0("Topic ", topic),
         term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill = as.factor(topic))) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y") +
  coord_flip() +
  scale_x_reordered() +
  labs(x = NULL, y = expression(beta),
       title = "Highest word probabilities for each topic",
       subtitle = "Words associated with each topic",
       caption = "Graphic: Isha Mahajan \n Fig 1.2")+
       scale_color_manual(aesthetics = "Darjeeling2")+
  theme_fivethirtyeight()
```

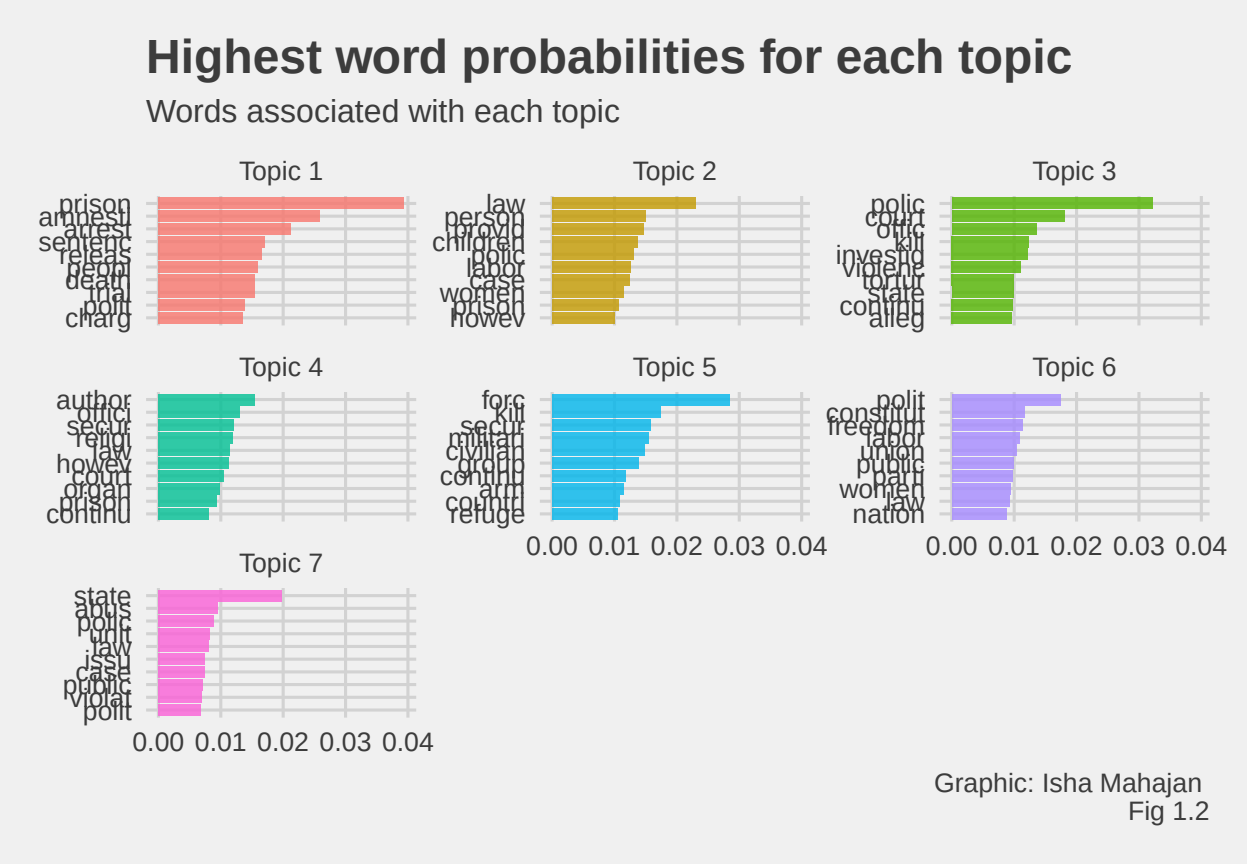


Fig 1.2

```
print(figeone_two)
```

```
## # A tibble: 5,278 x 3
##    topic term          beta
##    <int> <chr>        <dbl>
##  1     1 abl    0.000107
##  2     2 abl    0.000207
##  3     3 abl    0.0000121
##  4     4 abl    0.000705
##  5     5 abl    0.000484
##  6     6 abl    0.000717
##  7     7 abl    0.000607
##  8     1 abroad 0.000485
##  9     2 abroad 0.000000174
## 10     3 abroad 0.00000271
## # i 5,268 more rows
```

```
#ggsave(figeone_two, "fig1.2.png", dpi = 400)
```

```
metadata <- read_csv("dataverse_files_acled/reports_metadata.csv")
```

```
## Rows: 14190 Columns: 26
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr  (6): file_name, new_filename, country_iso3c, country_name, report_name,...
## dbl (20): year.0, word_count, hathaway, state, fariss.mean, fariss.std_devia...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
metadata_transformed <- metadata %>%
  group_by(organization, year.0) %>%
  summarise(count = n())
```

```
## `summarise()` has grouped output by 'organization'. You can override using the
## `.groups` argument.
```

```
figone_three <- ggplot(metadata_transformed, aes(x = year.0, y = count, color = organization)) +
  geom_line() +
  labs(title = "Number of Reports Published Each Year by Organization",
       x = "Year",
       y = "Report Count",
       legend = "Organization",
       caption = "Graphic: Isha Mahajan \n Fig 1.3") +
  theme_minimal() +
theme(
    title = element_text(face = "bold"),
    legend.title = element_text(face = "bold"),
    legend.box.background = element_rect(color = "black", linetype = "solid")
  )
print(figone_three)
```

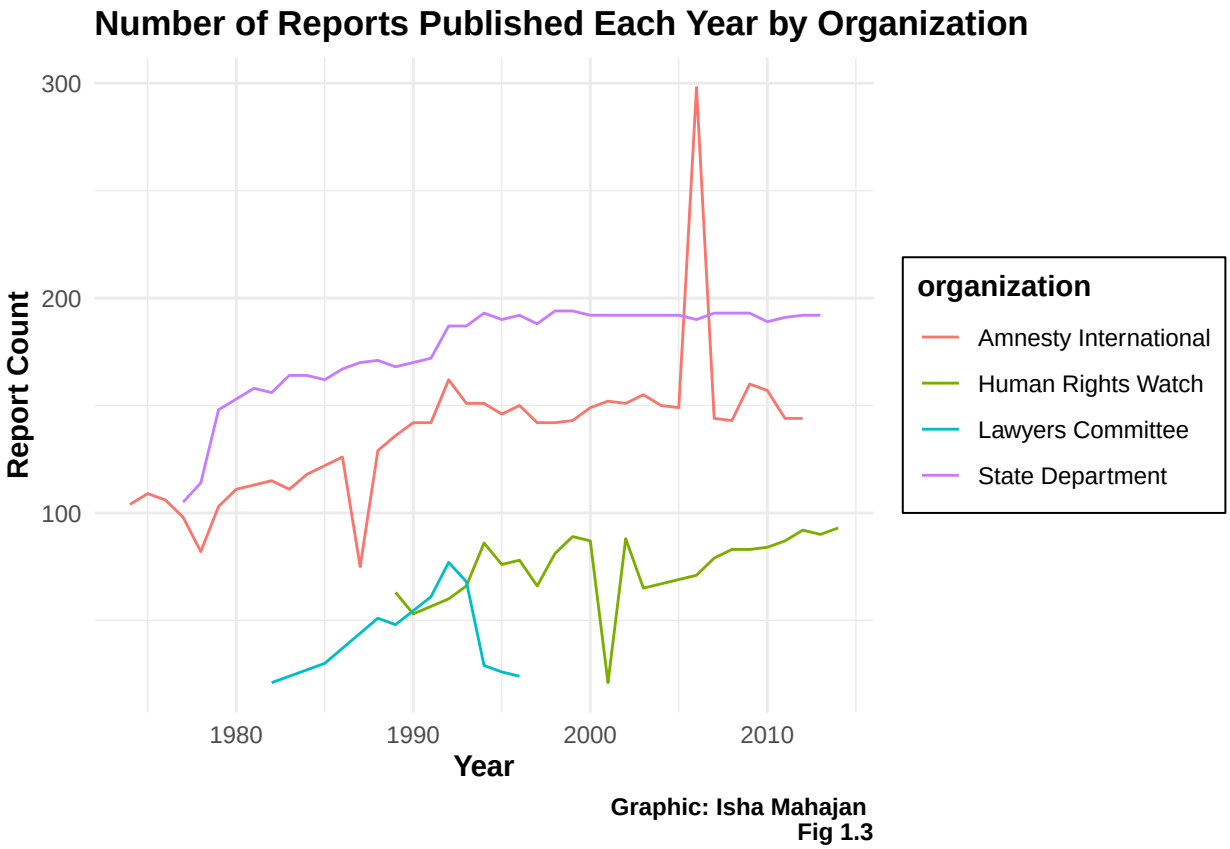


Fig 1.3

```
#ggsave(figone_three, "Fig1.3.png", dpi = 400)
```

Refelections (Part 3)

This is an initial analysis and exploration to build a structural topic model and explore the potential of using Natural Language Processing in the field of Human Rights. By looking at a random sample of 1000 documents, this model, with a short run time, was able to generate topics and probabilities of a document belonging to a certain topic. This can work in parallel with human coders who have to go through volumes of texts and generate thematic codes to classify them into categories. If iterated upon, a model like this can serve as a good starting point to automate some of those processes, and serve useful to organizations like ACLED to diversify their data sources by analyzing large volumes of texts and generating insights for the broader research community in political violence and global affairs.

Keeping this model at the core of building out a process, I would like to work with this at scale depending on the computing power available. By using popular libraries in R/Python like Beautiful Soup or API calls, we could leverage large volumes of text to train a model and generate initial topical insights. Thereafter,the model can serve as a starting point to share topical prevalence of documents from websites like amnesty, human rights, landmine monitor etc. to provide the research community an opportunity to make their search processes more streamlined, enable coders to work in tandem with the model to increase it's accuracy, and eventually scale this into a predictive model where we can predict the time when the conflict would we reported, classify the organization by which a text was published etc.

The key features of this tool would be:

1. Generating and Contextualizing topics from large volumes of text

2. Opportunity to select organizations who's text the user is interested in exploring; the ability to see the topical prevalence in those text
    3. Forecast whether a future report/document by these organizations would be classified into a certain topic or not.
        4. Forecast the time when a conflict would be reported and perhaps exploring the lag from time of conflict to time of reporting