

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 errors
```

```
library(quantda)
```

```
## 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 = 'min')
```

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

```
## - fewer than 75 times: 70,687
```



```

#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)

# 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

```

```

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

```

```

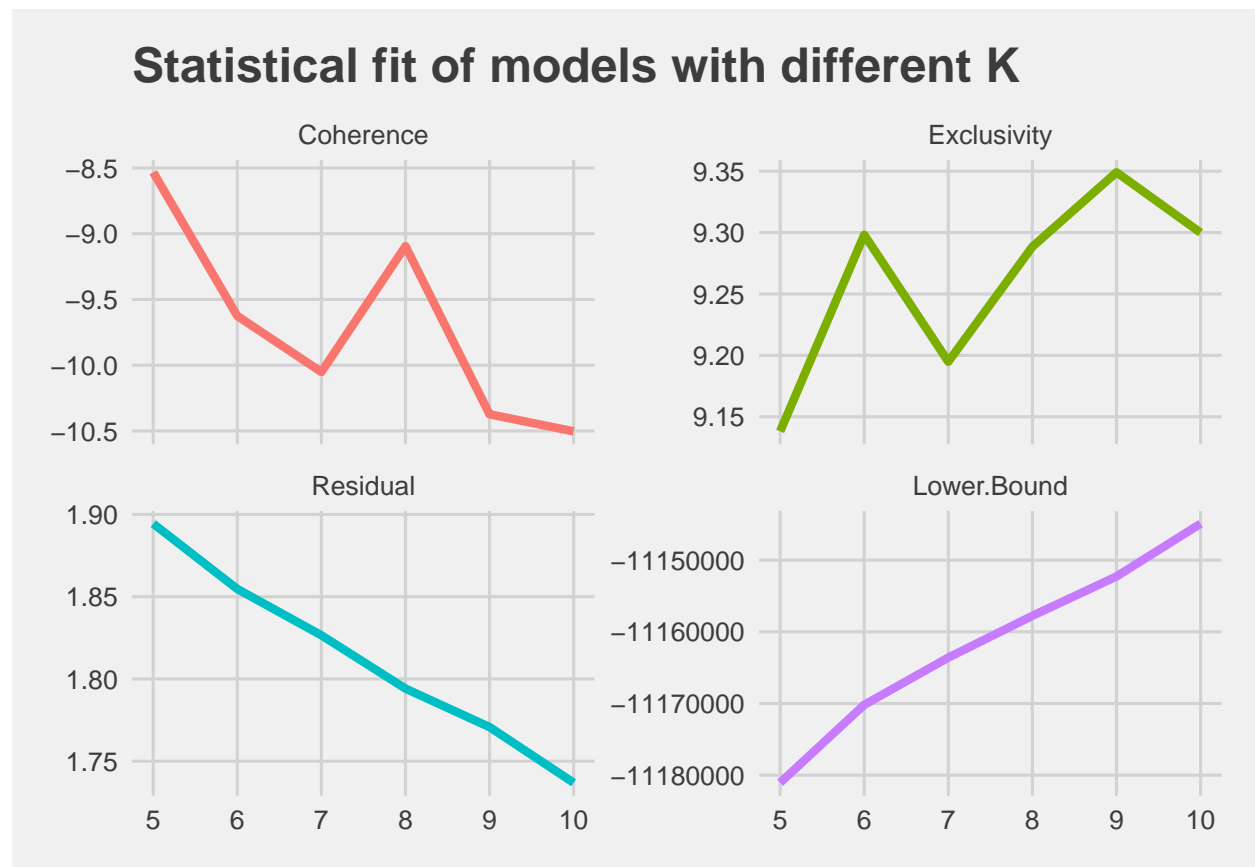
##      K      variable      value
## 1  5   Coherence -8.532884e+00
## 2  6   Coherence -9.625646e+00
## 3  7   Coherence -1.005178e+01
## 4  8   Coherence -9.095045e+00
## 5  9   Coherence -1.037362e+01
## 6 10   Coherence -1.050317e+01
## 7  5 Exclusivity  9.138114e+00
## 8  6 Exclusivity  9.298216e+00
## 9  7 Exclusivity  9.194833e+00
## 10 8 Exclusivity  9.288499e+00
## 11 9 Exclusivity  9.349108e+00
## 12 10 Exclusivity 9.299925e+00
## 13 5   Residual  1.894368e+00
## 14 6   Residual  1.854690e+00
## 15 7   Residual  1.826521e+00
## 16 8   Residual  1.794122e+00
## 17 9   Residual  1.770726e+00
## 18 10  Residual  1.736895e+00
## 19 5 Lower.Bound -1.118106e+07
## 20 6 Lower.Bound -1.117019e+07
## 21 7 Lower.Bound -1.116360e+07
## 22 8 Lower.Bound -1.115777e+07
## 23 9 Lower.Bound -1.115228e+07
## 24 10 Lower.Bound -1.114492e+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.
```

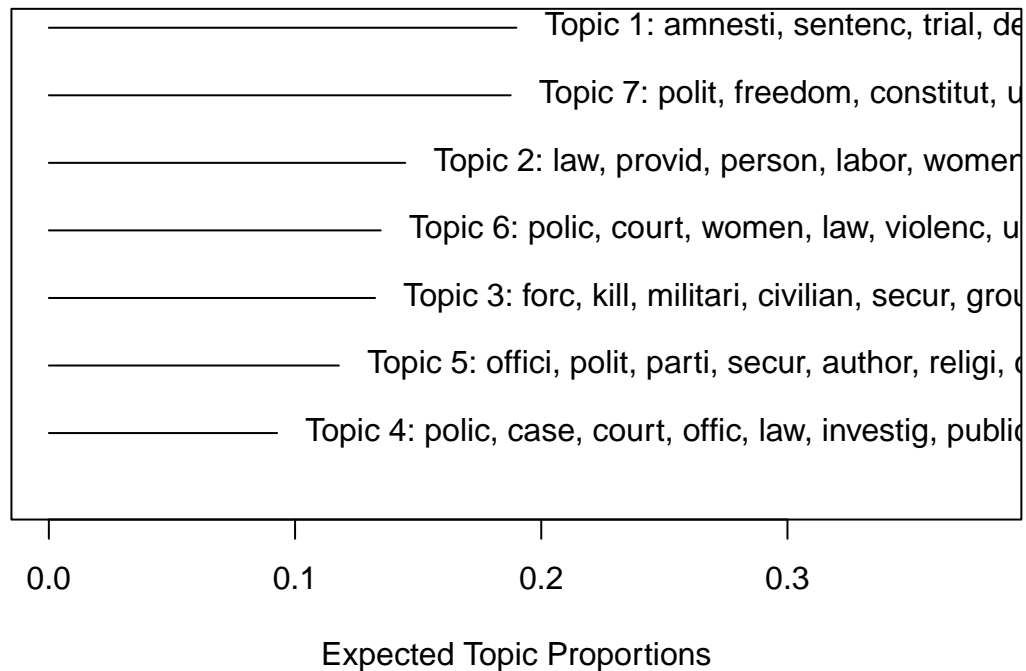
```
print(fit_stats)
```



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

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

Top Topics



```
# 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: amnesti, sentenc, trial, death, releas, peopl, polit
## FREX: amnesti, trial, sentenc, releas, death, execut, peopl
## Lift: amnesti, appar, inquiri, beaten, imprison, said, execut
## Score: amnesti, peopl, sentenc, death, releas, trial, tortur
## Topic 2 Top Words:
## Highest Prob: law, provid, person, labor, women, children, worker
## FREX: prohibit, labor, provid, child, children, employ, person
## Lift: workweek, disabl, child, prohibit, sexual, parent, age
## Score: workweek, labor, percent, traffick, sexual, child, women
## Topic 3 Top Words:
## Highest Prob: forc, kill, militari, civilian, secur, group, abus
## FREX: civilian, kill, war, attack, forc, militari, arm
## Lift: displac, war, soldier, civilian, humanitarian, conflict, armi
## Score: displac, kill, civilian, soldier, forc, militari, attack
## Topic 4 Top Words:
## Highest Prob: polic, case, court, offic, law, investig, public
## FREX: judg, prosecutor, offic, ministri, media, feder, case
## Lift: perform, prosecutor, judg, branch, dismiss, file, feder
```

```
##      Score: perform, prosecutor, traffick, polic, percent, offic, media
## Topic 5 Top Words:
##      Highest Prob: offici, polit, parti, secur, author, religi, continu
##      FREX: religi, opposit, foreign, restrict, ethnic, newspaper, offici
##      Lift: whose, muslim, permiss, religi, christian, ban, news
##      Score: whose, religi, see, foreign, section, opposit, activist
## Topic 6 Top Words:
##      Highest Prob: polic, court, women, law, violenc, un, penalti
##      FREX: un, penalti, violenc, convent, express, concern, crime
##      Lift: penalti, un, convent, risk, world, face, recommend
##      Score: penalti, un, women, peopl, sexual, violenc, death
## Topic 7 Top Words:
##      Highest Prob: polit, freedom, constitut, union, public, labor, law
##      FREX: respect, religion, freedom, constitut, grant, guarante, econom
##      Lift: guarante, economi, emigr, product, b, freeli, religion
##      Score: guarante, labor, percent, b, bargain, polit, freedom
```

```
#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	prohibit	civilian	judg	religi	un	respect
## 2	trial	labor	kill	prosecutor	opposit	penalti	religion
## 3	sentenc	provid	war	offic	foreign	violenc	freedom
## 4	releas	child	attack	ministri	restrict	convent	constitut
## 5	death	children	forc	media	ethnic	express	grant
## 6	execut	employ	militari	feder	newspap	concern	guarante
## 7	peopl	person	arm	case	offici	crime	econom
## 8	imprison	worker	displac	polic	parti	face	union
## 9	held	traffick	soldier	accord	muslim	crimin	c
## 10	tortur	age	armi	minor	allow	justic	free
## 11	detain	disabl	conflict	code	whose	women	b
## 12	detaine	discrimin	unit	station	ban	rule	wage
## 13	appeal	percent	abus	institut	meet	world	popul
## 14	alleg	practic	area	crimin	deni	fail	travel
## 15	charg	sexual	effort	suprem	activist	act	tradit
## 16	said	general	humanitarian	court	student	access	d
## 17	without	domest	peac	claim	demonstr	asylum	strike
## 18	custodi	minimum	camp	perform	see	head	percent
## 19	convict	enforc	target	investig	local	visit	assembl
## 20	appar	sector	depart	file	control	protect	labor
## 21	disappear	health	threat	problem	activ	polic	within
## 22	end	women	group	facil	detain	peopl	trade
## 23	detent	wage	secur	administr	section	sexual	practic
## 24	least	approxim	commit	requir	critic	commiss	polit
## 25	die	work	support	victim	leader	prosecut	press
## 26	other	law	near	note	permit	defend	economi
## 27	decemb	educ	region	attorney	regist	committe	privat


```
## 28   receiv   corrupt      develop   traffick   close recommend   speech
## 29     three     hour      respons    local     howev     death   social
## 30   beaten   problem          part     lack   permiss    violat     must
```

```
#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 0.0712894902 0.012488711 0.209481151 0.033737008 0.468680555
## 2:      2 0.0155450907 0.075315711 0.087149321 0.208130455 0.014461357
## 3:      3 0.0679017173 0.180873602 0.283737856 0.220091671 0.009655505
## 4:      4 0.6503225949 0.001698537 0.033448135 0.019034953 0.149353488
## 5:      5 0.0001853851 0.475251067 0.001540769 0.005538714 0.005257568
## 6:      6 0.0410834123 0.011135769 0.071816339 0.097419963 0.216824773
## 7:      7 0.0875552770 0.238561678 0.024190202 0.015754250 0.296512021
## 8:      8 0.0047100669 0.219503270 0.128828802 0.390373874 0.171487748
## 9:      9 0.5630891380 0.001737493 0.275045967 0.062486248 0.010324878
## 10:     10 0.0377045238 0.169340193 0.005666680 0.208093961 0.446304246
##      Topic6      Topic7
## 1: 0.045775196 0.158547889
## 2: 0.591686291 0.007711774
## 3: 0.007399700 0.230339948
## 4: 0.137458865 0.008683427
## 5: 0.000995231 0.511231266
## 6: 0.008432234 0.553287509
## 7: 0.031850210 0.305576361
## 8: 0.011713083 0.073383157
## 9: 0.078302030 0.009014246
## 10: 0.016009748 0.116880649
```

Visualization and Insights (Part 2 of the Technical Assignment)

In this section, I develop some exploratory graphs to see the topic prevalence in our sample. Fig 1.1 shows the proportion of topical prevalence 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 initial exploration point to observe keywords and see if there are thematic trends prevalent 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     0.0000887
## 2     2 abl     0.000167
## 3     3 abl     0.000333
## 4     4 abl     0.000355
## 5     5 abl     0.00104
## 6     6 abl     0.0000980
```

```
#convert 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      1 0.0713
## 2 2      2      1 0.0155
## 3 3      3      1 0.0679
## 4 4      4      1 0.650
## 5 5      5      1 0.000185
## 6 6      6      1 0.0411
## 7 7      7      1 0.0876
## 8 8      8      1 0.00471
## 9 9      9      1 0.563
## 10 10     1 0.0377
## # i 6,990 more rows
```

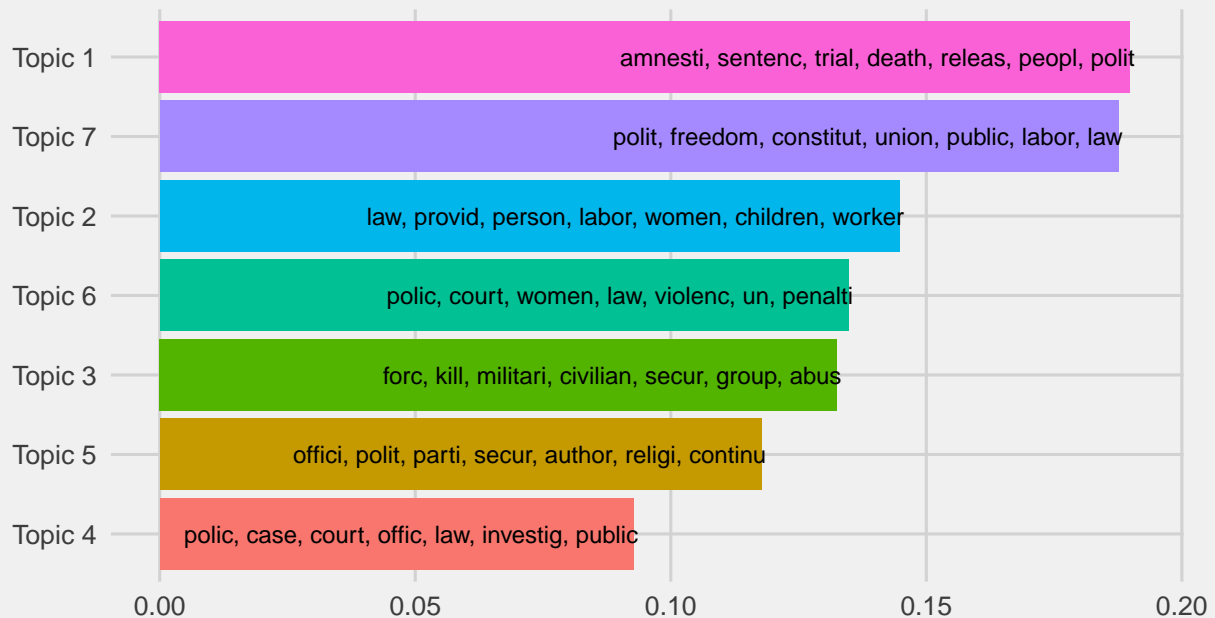
```
top_terms <- model_beta %>%
  arrange(beta) %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  arrange(-beta) %>%
  select(topic, term) %>%
  summarise(terms = list(term)) %>%
  mutate(terms = map(terms, paste, collapse = ", ")) %>%
  unnest(cols = c(terms))
```

```
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)
```

Top Seven Topics in Human Rights Texts

Seven topics by prevalence with the top words that contribute to each topic



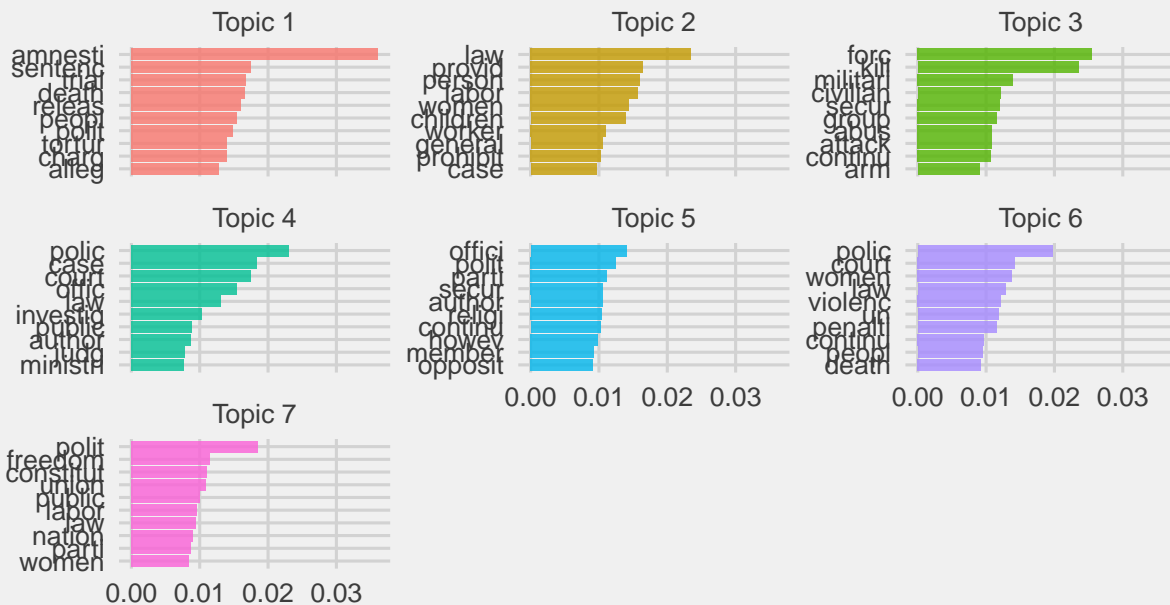
Graphic: Isha Mahajan
Fig 1.1

```
#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()
```

Highest word probabilities for each topic

Words associated with each topic



Graphic: Isha Mahajan
Fig 1.2

```
print(figeone_two)
```

```
## # A tibble: 5,264 x 3
##   topic term      beta
##   <int> <chr>    <dbl>
## 1     1  abl  0.0000887
## 2     2  abl  0.000167
## 3     3  abl  0.000333
## 4     4  abl  0.000355
## 5     5  abl  0.00104
## 6     6  abl  0.0000980
## 7     7  abl  0.000599
## 8     1 abroad 0.000276
## 9     2 abroad 0.00000310
## 10    3 abroad 0.0000524
## # i 5,254 more rows
```

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

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

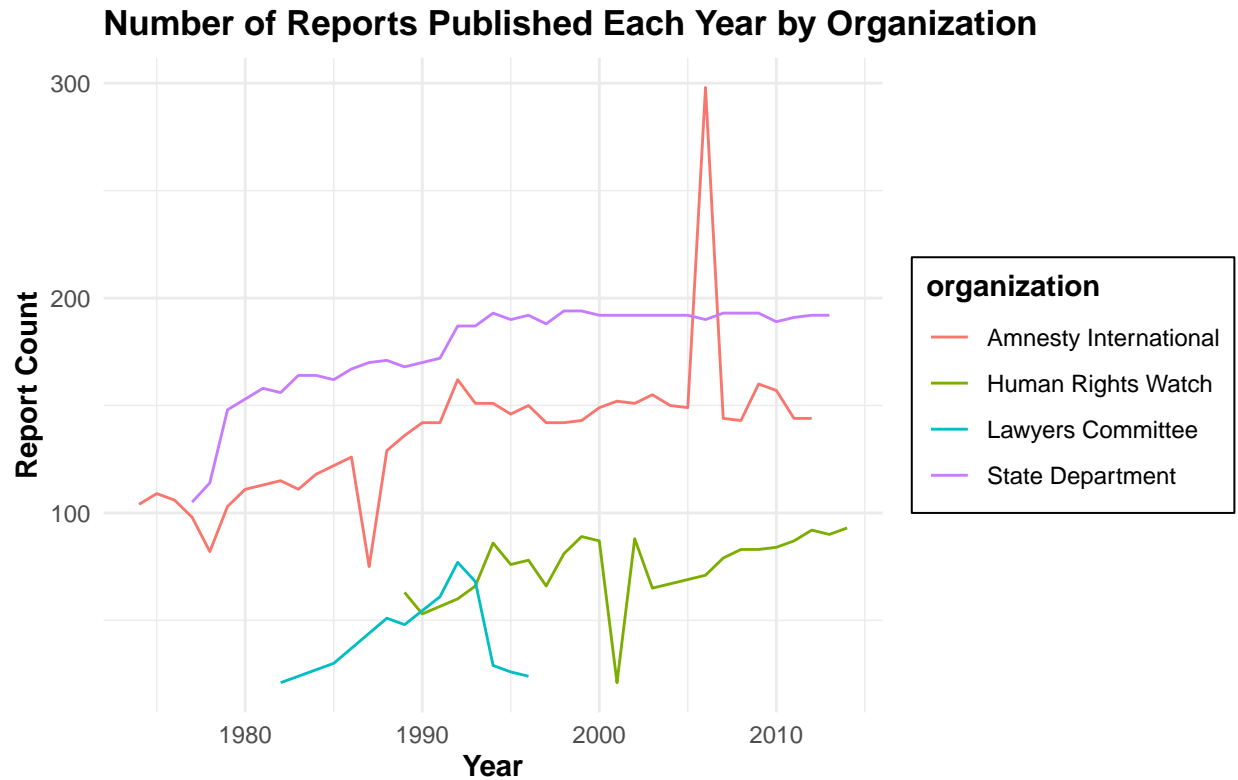
```
## Rows: 14190 Columns: 26
## -- Column specification -----
## Delimiter: ","
```

```
## 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)
```



Graphic: Isha Mahajan
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 its accuracy, and eventually scale this into a predictive model where we can predict the time when the conflict would be reported, classify the organization by which a text was published etc.

The key features of this tool would be:

Generating and Contextualizing topics from large volumes of text

Opportunity to select organizations who's text the user is interested in exploring; the ability to see the topical prevalence in those text

Forecast whether a future report/document by these organizations would be classified into a certain topic or not.

Forecast the time when a conflict would be reported and perhaps exploring the lag from time of conflict to time of reporting