Newspaper coverage of wars i the 21st century

Computational Social Science - 2022

Samuel Rauh

Introduction

The relationship between the media and wars has always been a controversial one. In such times it is very difficult to get down to the source and the large media companies are most of the time the only ones which have the ability to cover the state of a war since a whole region is in a state of emergency and it is almost impossible for normal citizens to get information otherwise. The media is in a critical position because they have the crucial ability to get the information and it is their determination to distribute it. But from the perspective of the media they are companies with a financial incentive. They earn a lot of money from international crises. The two motives of being a reliable source and attracting a large audience might contradict with each other. In addition have most large media groups a stronger or less strong political agenda which they push and which can have a severe impact on how they cover news. This can result in multiple extremely different portrayals of the same topic. This critical role of the media in times of war has motivated me to have a closer look at how different media sources portray different wars.

Methods

To collect the data I use the "The Guardian" API of the British newspaper and the "New York Times" API of the American Newspaper. After cleaning up the data I will perform three text analysis procedures. First a sentiment analysis to check if there are differences in the positivity score between the wars but also between the two newspapers. Second I will do a word frequency analysis and look at the most used words. And third I will use the " Lexicoder policy agendas" dictionary to do a topic analysis. I chose to look at three different wars: The war in Afghanistan, the war going on right now in the Ukraine and the civil war in Syria. I have chosen those three because they are all different from each other in the point of how the "West" relates to them. The war in Afghanistan was driven by the USA following the 9/11 terrorist attack in New York. The war in Ukraine is geographically and also politically closer to the "West" but western forces have (not yet) intervened themselves. The war in Syria has been going on for a long time and has had a lot of media coverage in the past, it is geographically and culturaly much further away from the "West" but the USA and its allies also have played a controversial role in it.

Limitations

The primary limitation I had to set were the numbers of newspaper and wars I included in the analysis. I initially wanted to also use the "Die Zeit" API to have a newspaper in a different language and from another different country but the service has been stopped due to technical difficulties. Since there are only limited APIs I chose to stick to these two. But it would have been very interesting to also have a newspaper with a potentially right leaning political orientation since these two are both more liberal. I limited myself to those three wars since I don't think that it would make much of a difference to look at more wars. I think that differences in the reporting of the wars would show when comparing these three. For further studies it would be interesting to look at wars from longer time ago. I'm sure there would be interesting finding when looking at the Vietnam war or even the 2. world war.

Data Collection and Processing

Not all code chunks are run or shown, this is for the reason of illustration and very long time of gathering the data. All the data used is saved in the "Data" folder,

Libraires

```
library(dplyr)
library(quanteda)
library(jsonlite)
library(stringr)
library(tidyverse)
library(ggplot2)
library(rvest)
library(stringr)
library(quanteda.textplots)
```

The Guardian

URL generator for the Guardian API

```
guardian_key = read_lines("guardian_key.txt")

guardian_url = function(search_word, date_from='', date_to='') {
  search_word <- str_replace(search_word, ' ', '%20')

  if (date_from == '' | date_to == '') {
    url <- paste0('http://content.guardianapis.com/search?q=', search_word,
                  '&show-blocks=all&api-key=', guardian_key, sep='')
  } else {
    url <- paste0('http://content.guardianapis.com/search?q=', search_word,
                  '&from-date=', date_from, '&to-date=', date_to,
                  '&show-blocks=all&api-key=', guardian_key, sep='')
  }
  url
}
```

Afghanistan War

```
#There are 63'723 articles which is too much, it exceeds the rate limit of the Guardian API
#that's why I will extract 10% of all the pages of each relevant year.

afgh_list <- vector("list")
counter <- 1

for (year in 1999:2022) {
  base_url <- guardian_url('afghanistan war', paste0(year, '-01-01'),
                          paste0(year, '-12-31'))
  n_results <- fromJSON(base_url) %>% .$response %>% .$total
  max_pages <- ceiling(((n_results[1] / 10)-1))
  random_pages <- sample.int(max_pages, max_pages/10)
  print(year)
```

```
  for(i in 1:(max_pages/10)){
    x=random_pages[i]
    print(i)
    url <- paste0(base_url, "&page=", x, sep='')
      GuardSearch <- fromJSON(url, flatten = TRUE) %>%
        data.frame(.,stringsAsFactors = FALSE)
    afgh_list[[counter]] <- GuardSearch
    counter <- counter + 1
    #Sys.sleep(1)
}
}

afgh_results <- rbind_pages(afgh_list)

save(afgh_results, file = "Data/afgh_results.RData")
```

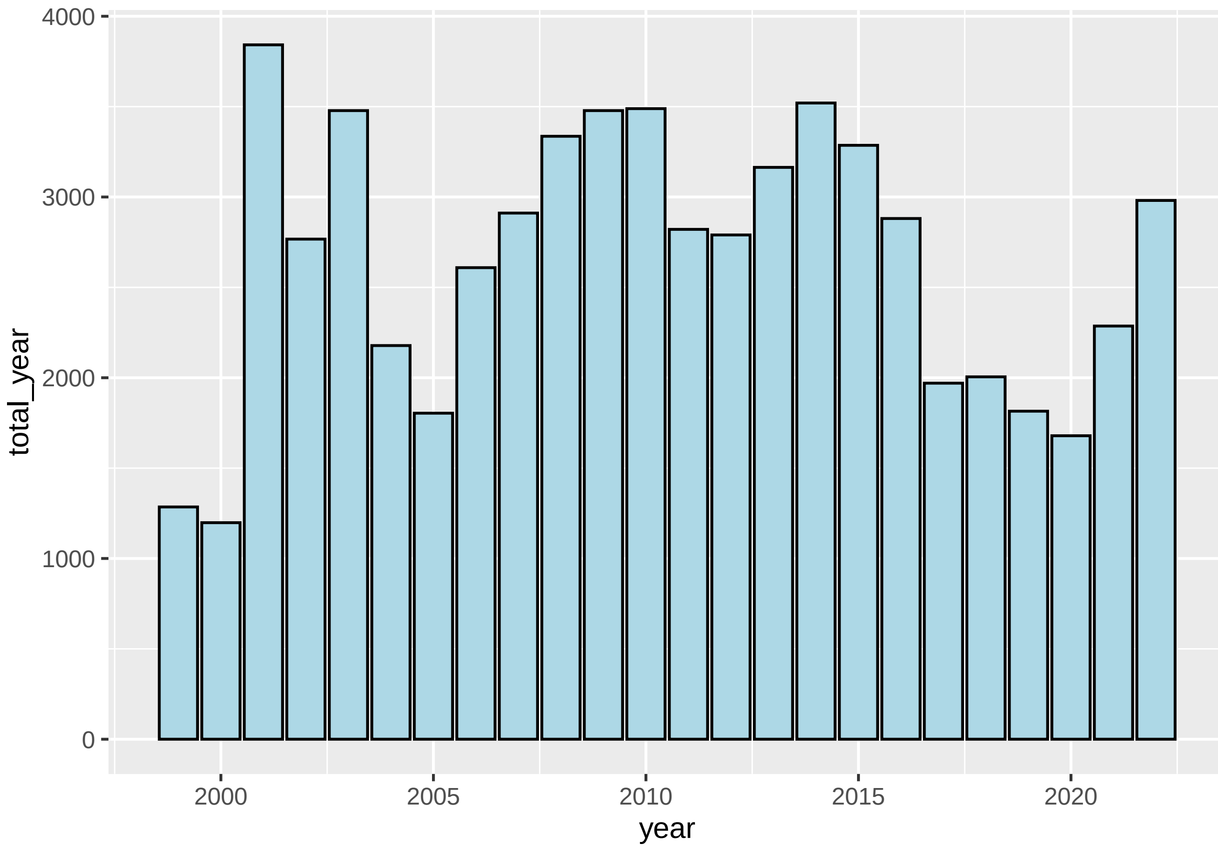
Get Data

```
load('Data/afgh_results.RData')

#Pick filter out the results of the type "article"
afgh_results <- afgh_results %>%
  filter(response.results.type == "article") %>%
  unnest(cols = response.results.blocks.body)

# Make a selection of the important variables

afgh <- afgh_results %>%
  select(id = response.results.id,
         url = response.results.webUrl,
         title = response.results.webTitle,
         text = bodyTextSummary,
         date = response.results.webPublicationDate,
         section = response.results.sectionName,
         total_year = response.total
         )

afgh$text_len <- nchar(afgh$text)
summary(afgh$text_len)

afgh$text[afgh$text_len < 50]

#filter out articles with no significant text
afgh <- afgh %>%
  filter(!text %in% c("", " ", "."))
summary(afgh$text_len)

#extract date and year

afgh$date <- gsub("T.*",
                  "",
                  afgh$date) %>% as.Date()
```

```r
afgh$year <- afgh$date %>%
  gsub("([0-9]{4}).*",
       "\\1", .) %>% as.numeric


afgh$total_year <- afgh$total_year %>%
  as.numeric

save(afgh, file = "Data/afgh.RData")
```

Data Cleaning

```r
# Create a histogram to show the distribution of articles

load("Data/afgh.RData")
afgh_hist <- ggplot(afgh, aes(x=date))+
  geom_histogram(binwidth = 200, fill='lightblue', color="black")+
  scale_x_date(date_breaks = "3 years", date_labels = "%Y")+
  theme_classic()


afgh_hist_year <- afgh[!duplicated(afgh[ , c("year")]), ] %>%
  select(total_year, year) %>%
  ggplot(aes(x=year, y=total_year))+
  geom_bar(stat='identity', fill = "lightblue", color="black")
afgh_hist_year
```



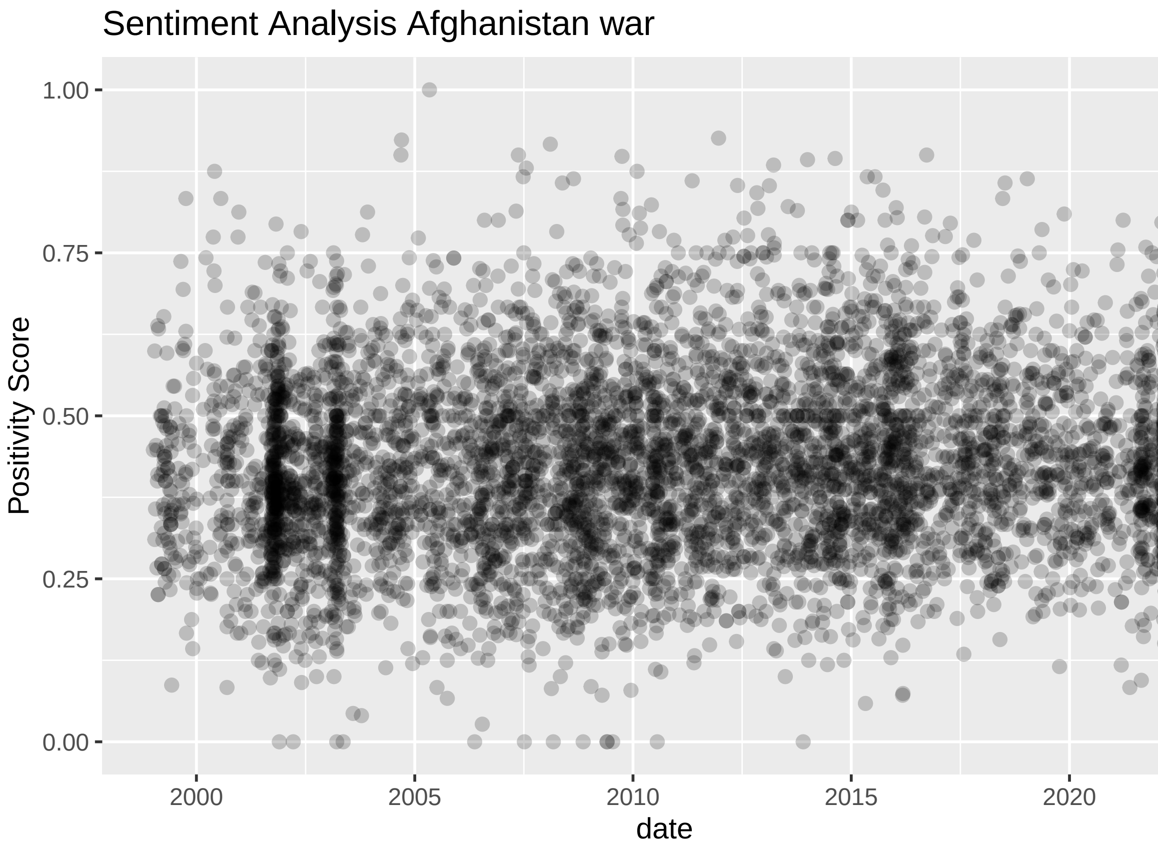
Basic Analysis

```
afgh_corpus <- corpus(afgh,
                      text_field = "text")

summary(afgh_corpus) %>% head
docvars(afgh_corpus) %>%
  head

afgh_corpus[1]


afgh_toks <- tokens(afgh_corpus,
                    what = c("word"),
                    remove_separators = TRUE,
                    include_docvars = TRUE,
                    ngrams = 1L,
                    remove_numbers = FALSE,
                    remove_punct = TRUE,
                    remove_symbols = FALSE,
                    remove_hyphens = FALSE)

afgh_toks %>% head

afgh_toks <- afgh_toks %>%
  tokens_tolower %>%
```

```r
  tokens_remove(stopwords("english"), padding = TRUE) %>%
  tokens_remove("") %>%
  tokens_wordstem(language = "english")

save(afgh_toks, file = "Data/afgh_toks.RData")
```

Create a corpus and tokens

```r
load("Data/afgh_toks.RData")
load("Data/afgh.RData")

# make a sentiment analysis on the ratio of positive to negative words in each article

afgh_toks_sent <- tokens_lookup(afgh_toks,
                                dictionary =  data_dictionary_LSD2015[1:2])
#afgh_toks_sent %>% head

afgh_dfm_sent <- dfm(afgh_toks_sent)
#afgh_dfm_sent %>% head

afgh_pos_neg <- afgh_dfm_sent %>% convert(.,to = "data.frame") %>%
  mutate(pos_to_neg = (positive / (positive + negative)))

#summary(afgh_pos_neg)
# afgh_pos_neg %>% filter(is.na(afgh_pos_neg$pos_to_neg))
# 1 NA in year 2002
# Row has 0 negative and 0 positive -> NaN -> discard

afgh_sent <- afgh

afgh_sent$pos_neg <- afgh_pos_neg$pos_to_neg
afgh_sent <- afgh_sent %>% filter(!is.na(afgh_sent$pos_neg))

# create plots to show the results

plot_afgh_sent <- ggplot(afgh_sent, aes(x=date, y=pos_neg))+
  geom_point(size=2, alpha=0.2)+
  labs(title = "Sentiment Analysis Afghanistan war", y="Positivity Score")
plot_afgh_sent
```



Sentiment Analysis

```
afgh_by_year <- afgh_sent$pos_neg %>%
  aggregate(by=list(afgh_sent$year), FUN = mean) %>% rename(year = Group.1,
                                                       pos_neg = x)
# afgh_by_year

plot_afgh_pos_by_year <- ggplot(afgh_by_year, aes(x=year, y=pos_neg))+
  geom_line(color="royalblue", size=1)+
  theme_light()+
  ylim(0.35, 0.5)+
  labs(y="Positivity Score", title = "Sentiment Analysis Afghanistan war by year")
plot_afgh_pos_by_year
```

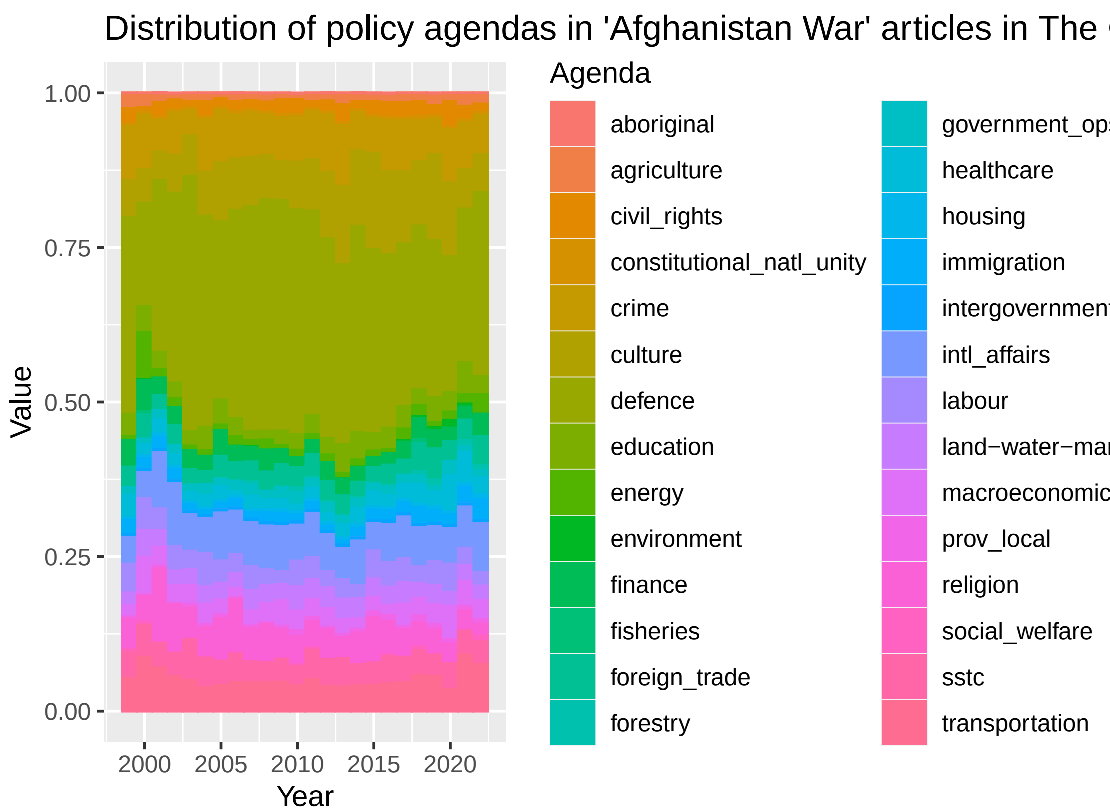

```
load("Data/afgh_toks.RData")


# load list with the most common words in English to remove them from the tokens

common_words <- read.delim("Data/1-1000.txt", header = FALSE) %>%
  head(200) %>%
  as.vector()
common_words <- common_words$V1

# removing the top 200 words from the tokens
afgh_toks_wordcount <- afgh_toks %>%
  tokens_remove(common_words)

afgh_dfm <- dfm(afgh_toks_wordcount)



afgh_topwords <- topfeatures(afgh_dfm, 50) %>%
  data.frame(word=names(.),
             frequency =.,
             row.names = c())

afgh_topwords
```

Word Frequency

```
##           word frequency
## 1          war     18408
## 2        peopl     10946
## 3       govern      8527
## 4      countri      7712
## 5  afghanistan      7167
## 6         last      7144
## 7         forc      6867
## 8         iraq      6318
## 9     militari      6204
## 10        mani      5707
## 11     british      5686
## 12      report      5547
## 13       state      5449
## 14      nation      5022
## 15       polit      4989
## 16      presid      4916
## 17        week      4889
## 18      attack      4888
## 19    american      4786
## 20        kill      4672
## 21     support      4561
## 22       secur      4350
## 23      includ      4306
## 24      intern      4294
## 25        sinc      4041
## 26     taliban      4006
## 27       still      3947
## 28      minist      3931
## 29       month      3852
## 30      public      3829
## 31          mr      3804
## 32       group      3760
## 33       offic      3746
## 34        told      3694
## 35       power      3568
## 36       troop      3540
## 37     soldier      3463
## 38     britain      3443
## 39      offici      3366
## 40      famili      3346
## 41     foreign      3302
## 42          uk      3286
## 43      leader      3280
## 44       anoth      3243
## 45         tri      3158
## 46       women      3139
## 47       chang      3112
## 48        hous      3111
## 49      believ      3075
## 50       fight      3074
```

```r
save(afgh_dfm, file = "Data/afgh_dfm.RData")
```

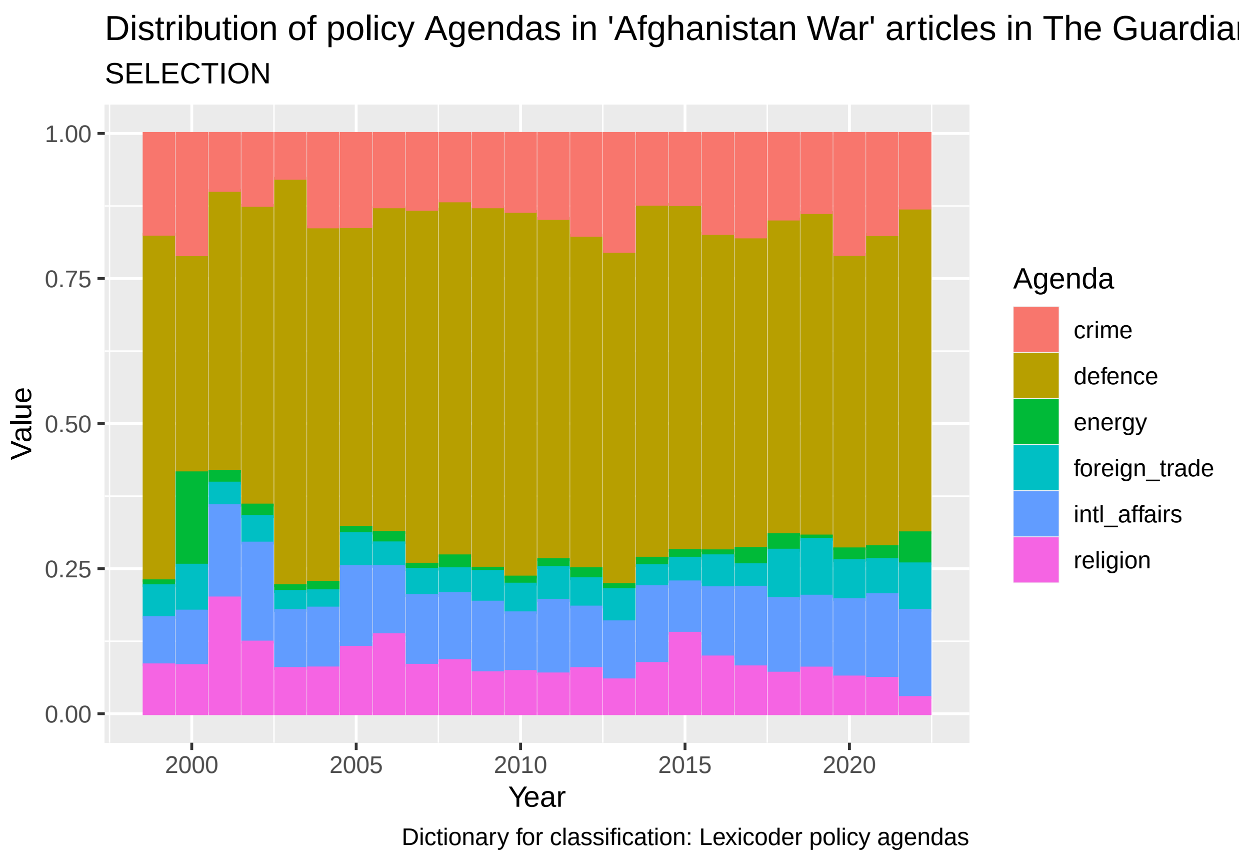
```r

load("Data/afgh_toks.RData")
load("Data/afgh.RData")

# Load the Lexicoder policy agendas
policyagendas <- dictionary(file = "Data/policy_agendas_english.lcd")

# lookup the policy agendas dictionary and give each article a score
afgh_toks_pol <- tokens_lookup(afgh_toks, dictionary = policyagendas)
afgh_pol <- dfm(afgh_toks_pol) %>%
  convert(to = "data.frame") %>%
  select(-doc_id)

# divide the values for each row through the sum of each row to get relative values for the agendas for
afgh_pol <- afgh_pol / rowSums(afgh_pol)

afgh_pol$year <- afgh$year

afgh_pol <- drop_na(afgh_pol)

# group by year
afgh_pol_by_year <- afgh_pol %>%
  group_by(year) %>%
  summarise_each(funs = sum)

# Plot the results to better inspect them
afgh_pol_by_year_plot1 <- afgh_pol_by_year %>%
  pivot_longer(cols = 2:29, names_to = "Agenda") %>%
  ggplot(aes(x=year, y=value, colour = Agenda, fill = Agenda))+
  geom_bar(position="fill", stat="identity")+
  labs(x="Year", y="Value", title="Distribution of policy agendas in 'Afghanistan War' articles in The G
       caption = "Dictionary for classification: Lexicoder policy agendas")
afgh_pol_by_year_plot1
```



ınary for classification: Lexicoder policy agendas

Policy agendas analysis

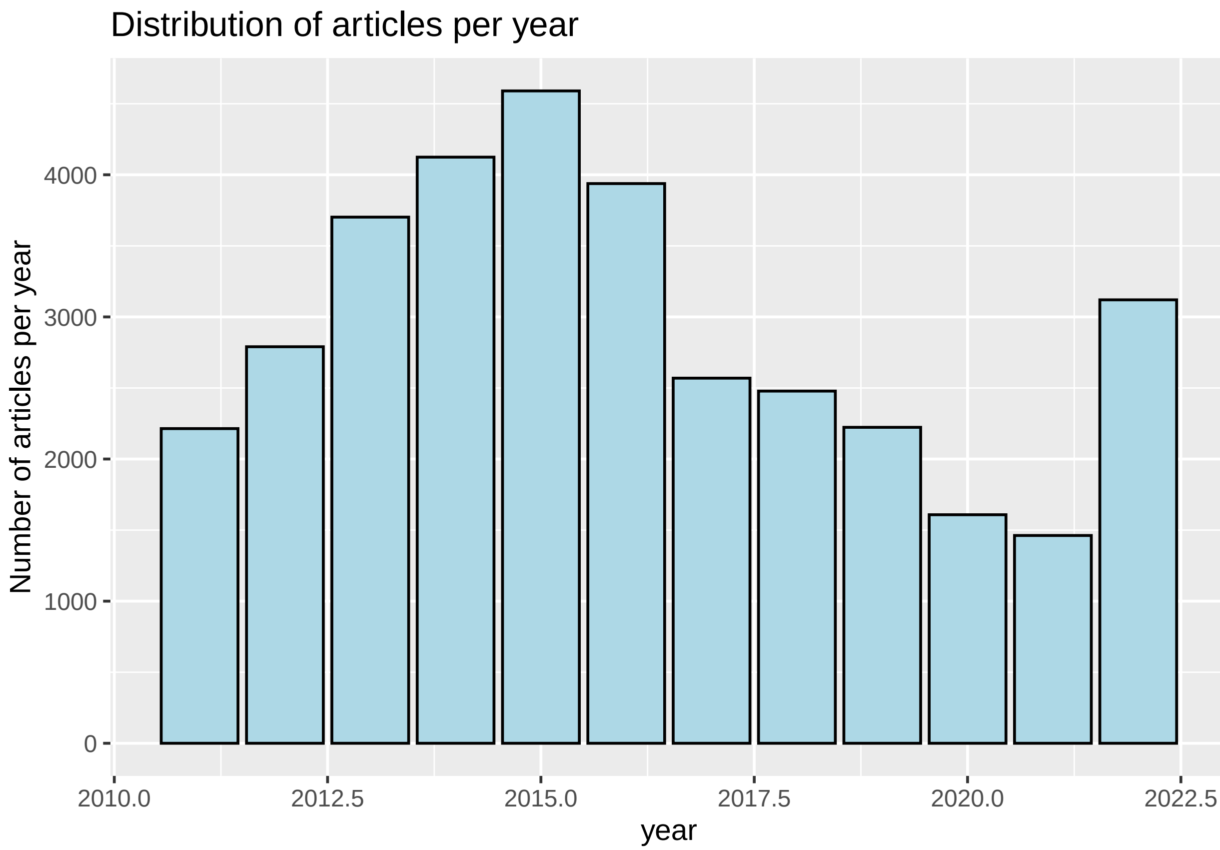
```r
# select some agendas which seam important
afgh_pol_by_year_plot_select <- afgh_pol_by_year %>%
  select(year, defence, energy, foreign_trade, intl_affairs, religion, crime) %>%
  pivot_longer(cols = 2:7, names_to = "Agenda") %>%
  ggplot(aes(x=year, y=value, colour = Agenda, fill = Agenda))+
  geom_bar(position="fill", stat="identity")+
  labs(x="Year", y="Value", title="Distribution of policy Agendas in 'Afghanistan War' articles in The (
       subtitle = "SELECTION", caption = "Dictionary for classification: Lexicoder policy agendas")
afgh_pol_by_year_plot_select
```



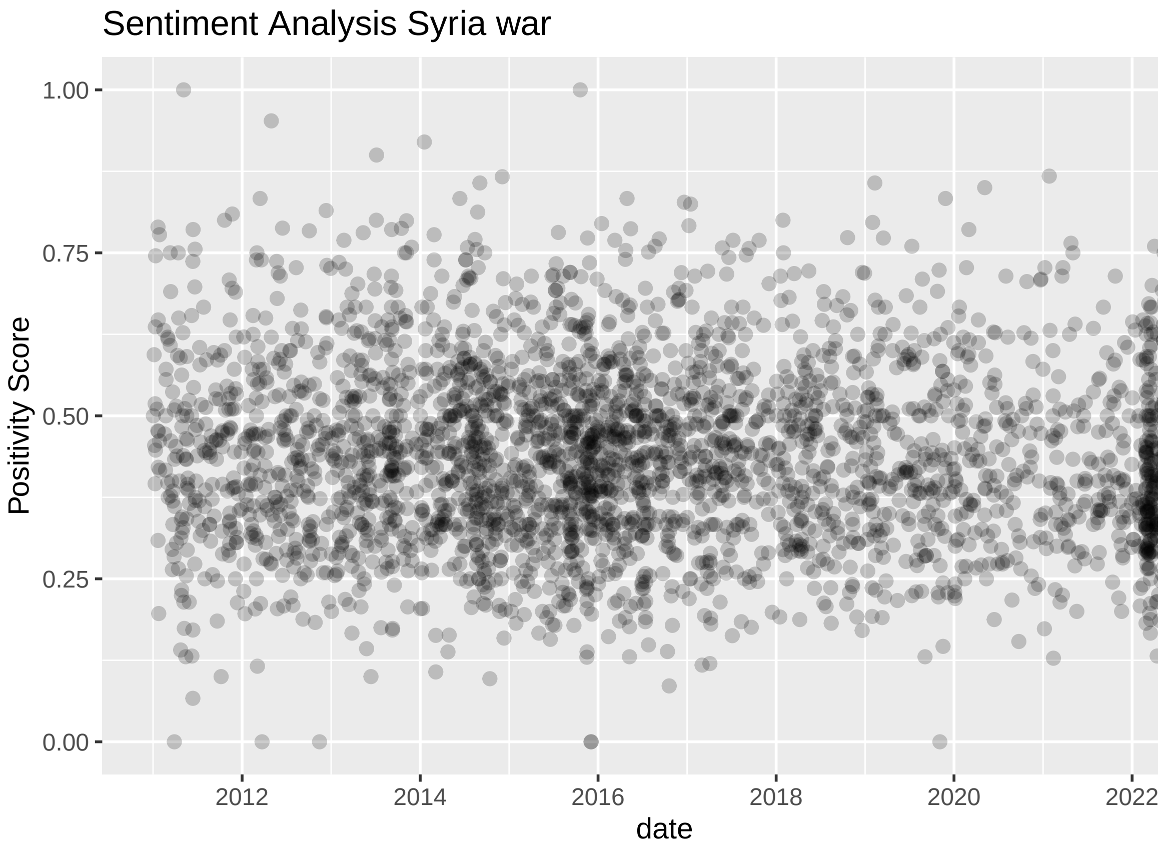

```
save(afgh_pol, file = "Data/afgh_pol.RData")
```

The code for the other wars are not included in the rendered markdown since they are almost identical to the coverage of the Afghanistan War

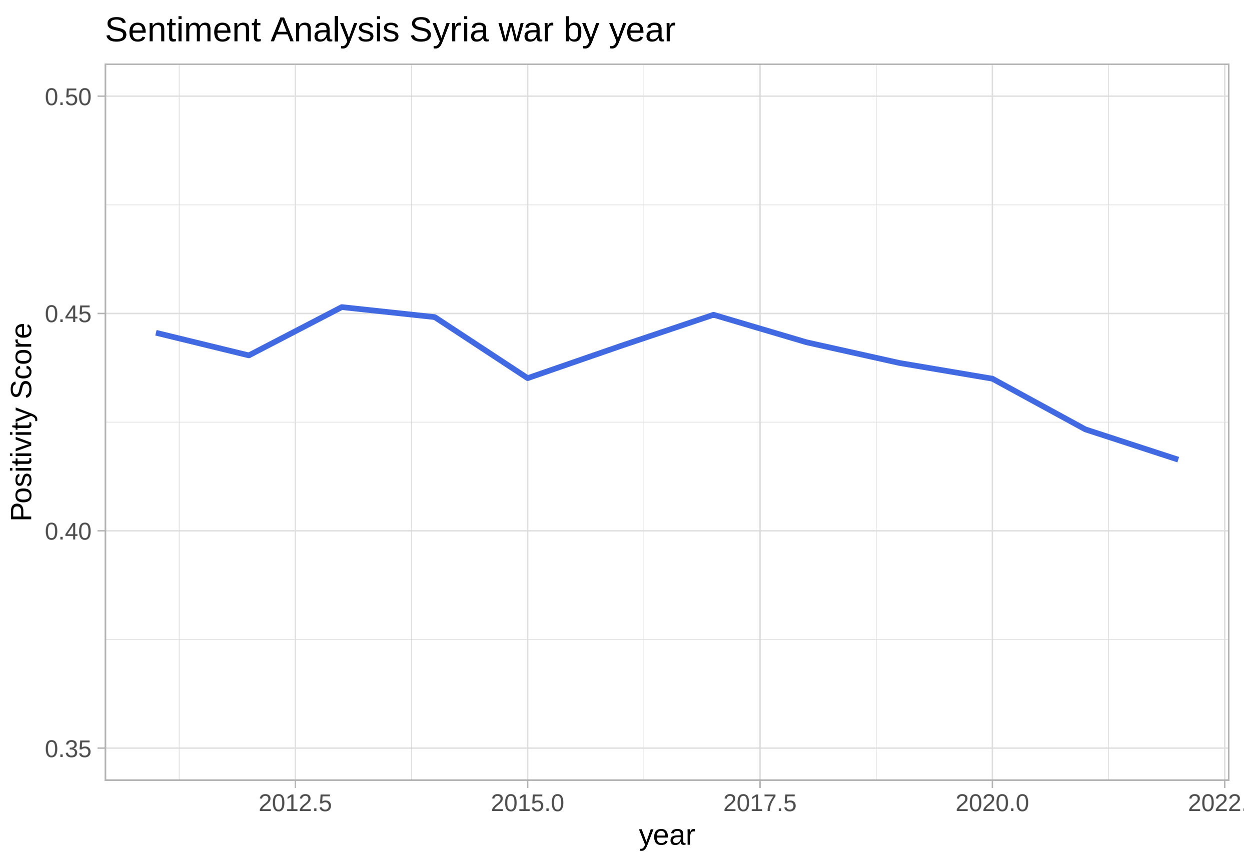
Syria War



Basic Analysis

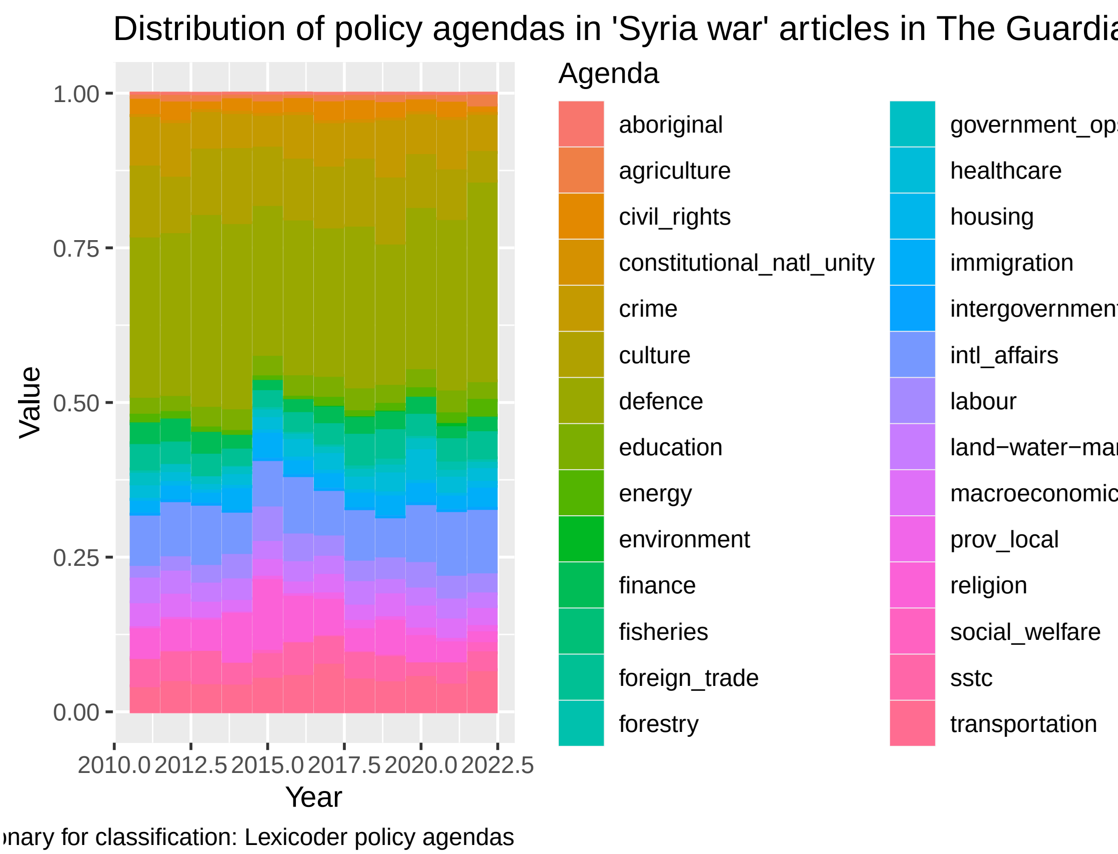


Sentiment Analysis

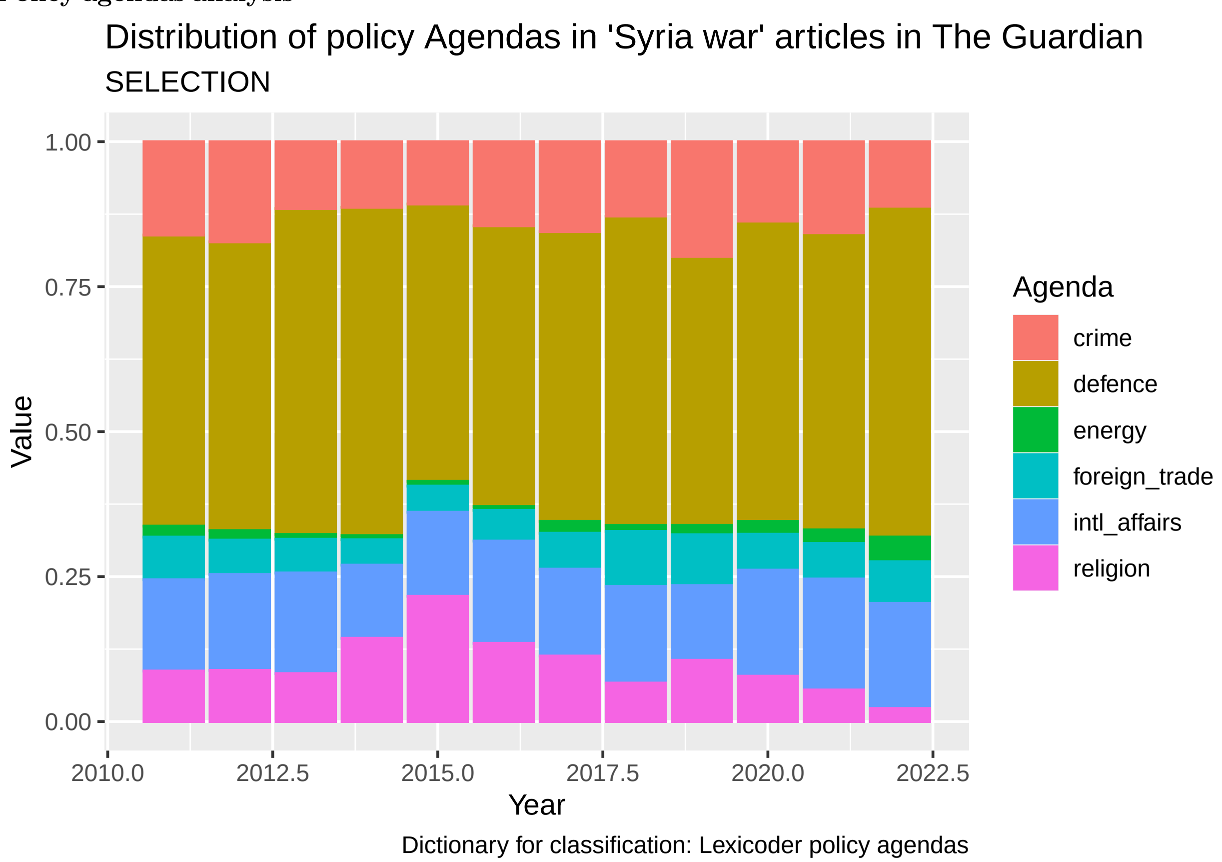


Word Frequency

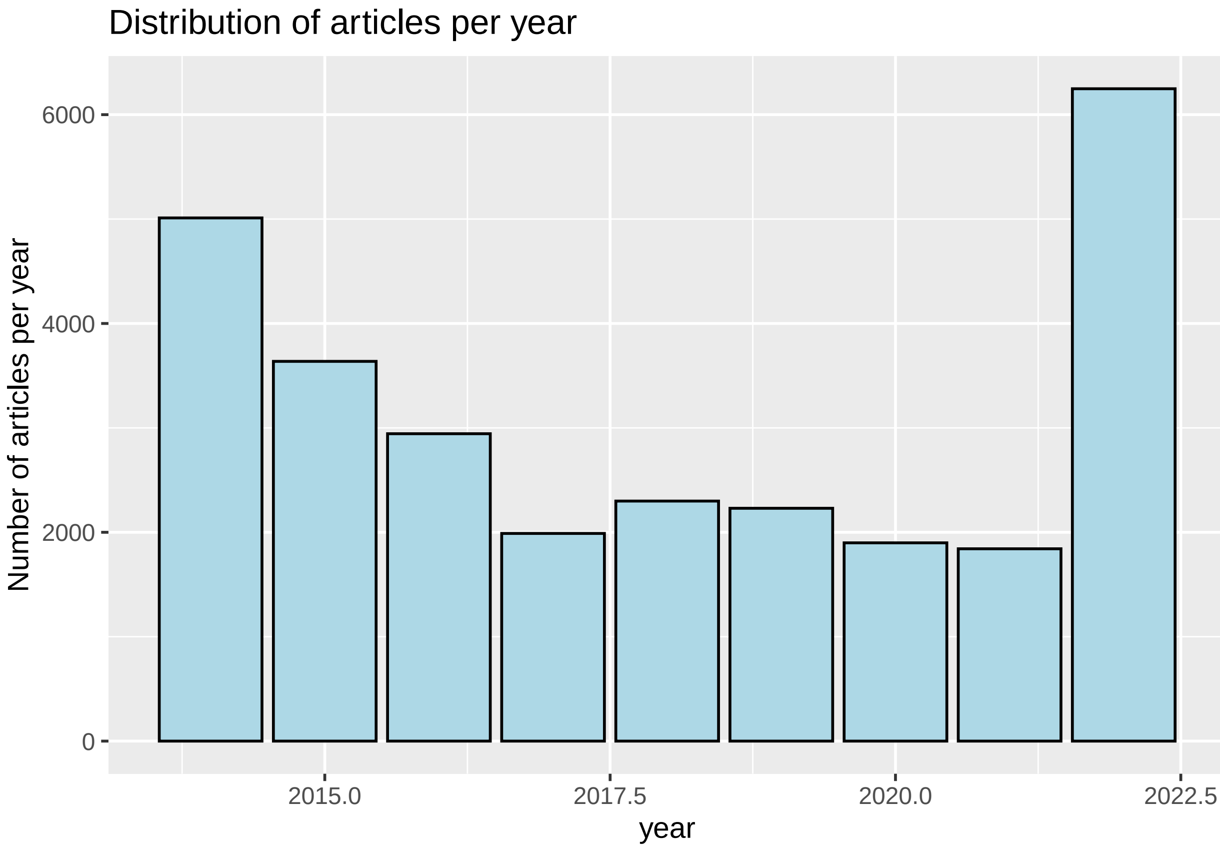
```
##        word frequency
## 1       war      8697
## 2     peopl      6562
## 3   countri      4552
## 4    govern      4493
## 5     syria      4228
## 6     state      3627
## 7      forc      3404
## 8      last      3367
## 9      mani      3016
## 10   report      2920
## 11 militari      2903
## 12    polit      2714
## 13   attack      2687
## 14   nation      2678
## 15   presid      2640
## 16     week      2634
## 17  support      2619
## 18   russia      2617
## 19   includ      2556
## 20   syrian      2466
## 21  russian      2410
## 22    group      2390
## 23     kill      2370
## 24    trump      2290
## 25       uk      2276
## 26     sinc      2275
## 27    secur      2262
## 28   intern      2246
## 29   famili      2229
## 30     iraq      2146
## 31     citi      2096
## 32     told      2092
## 33    still      2080
## 34    month      2077
## 35   minist      2038
## 36  british      1985
## 37   refuge      1914
## 38    power      1901
## 39      isi      1850
## 40    chang      1823
## 41 children      1794
## 42  foreign      1788
## 43    fight      1788
## 44   leader      1733
## 45   public      1732
## 46 conflict      1704
## 47   offici      1697
## 48   ukrain      1687
## 49      tri      1679
## 50    anoth      1658
```



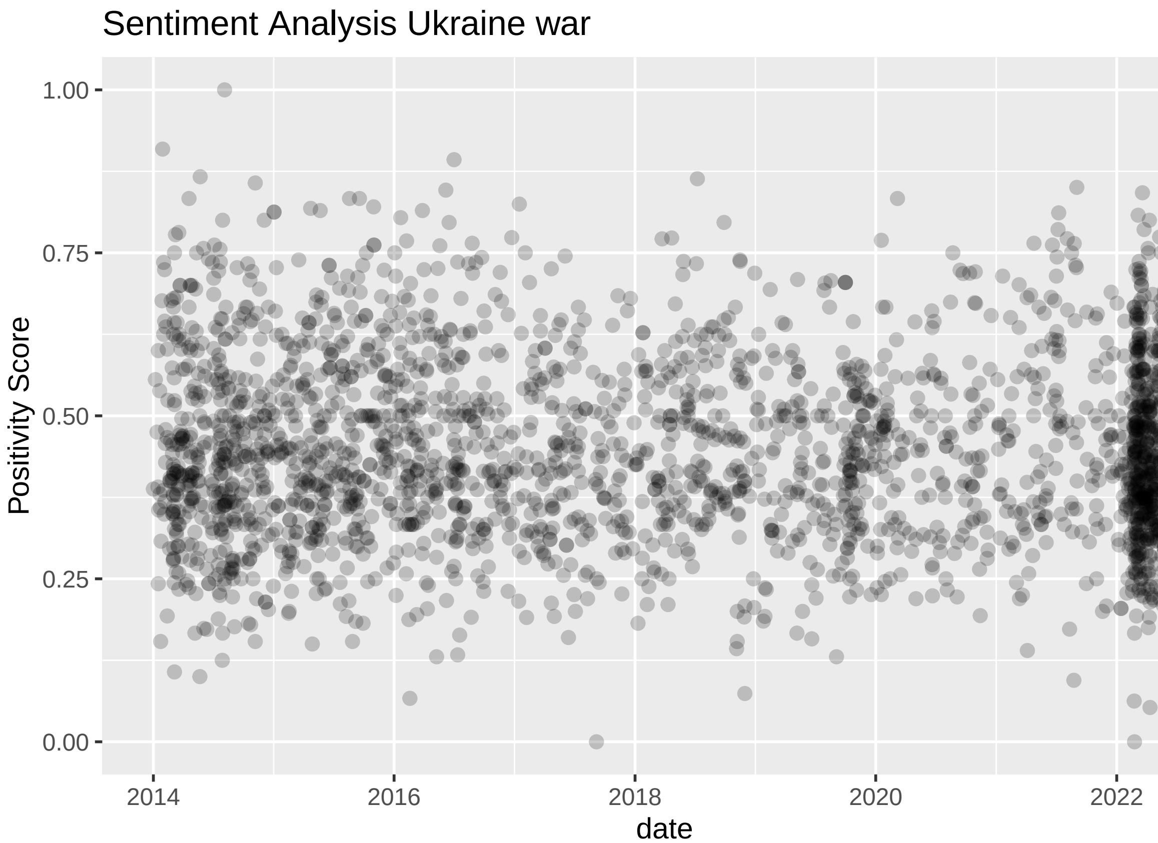
Policy agendas analysis



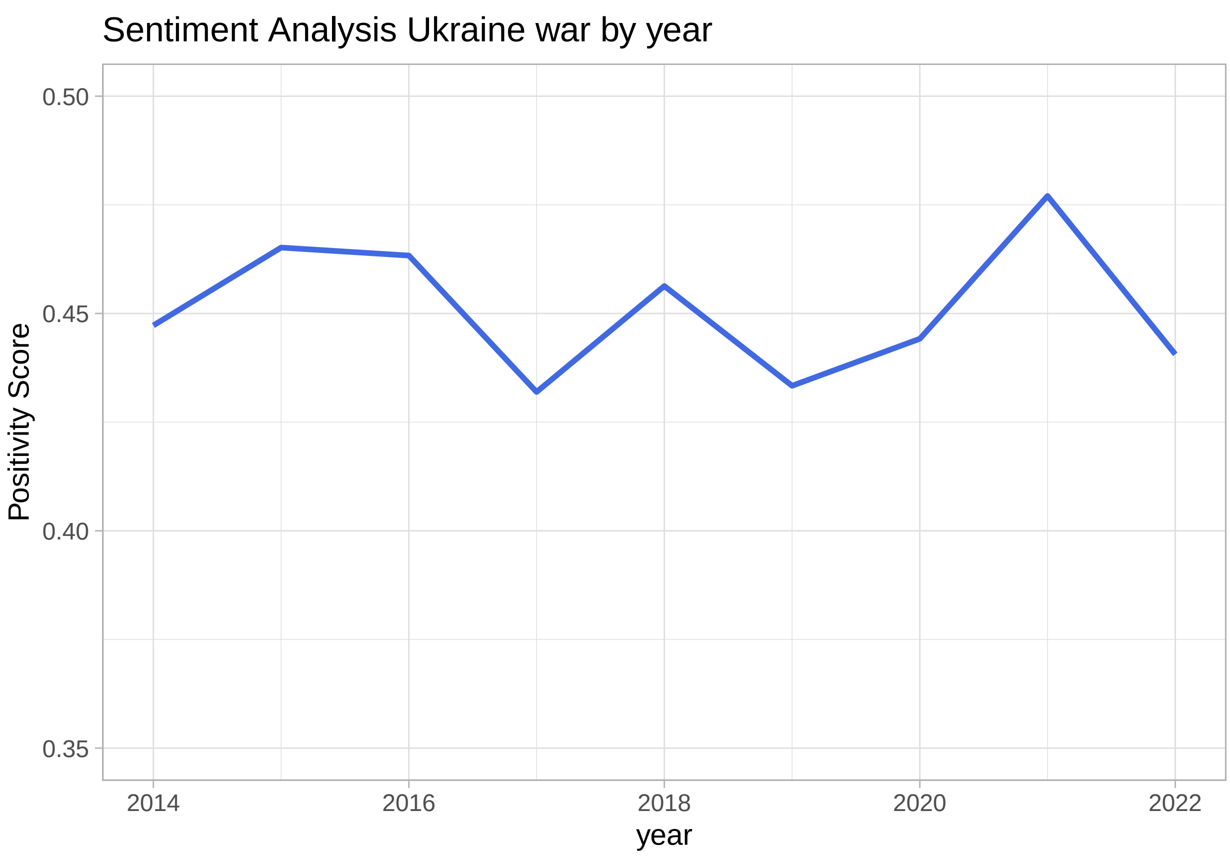
Ukraine War



Basic Analysis

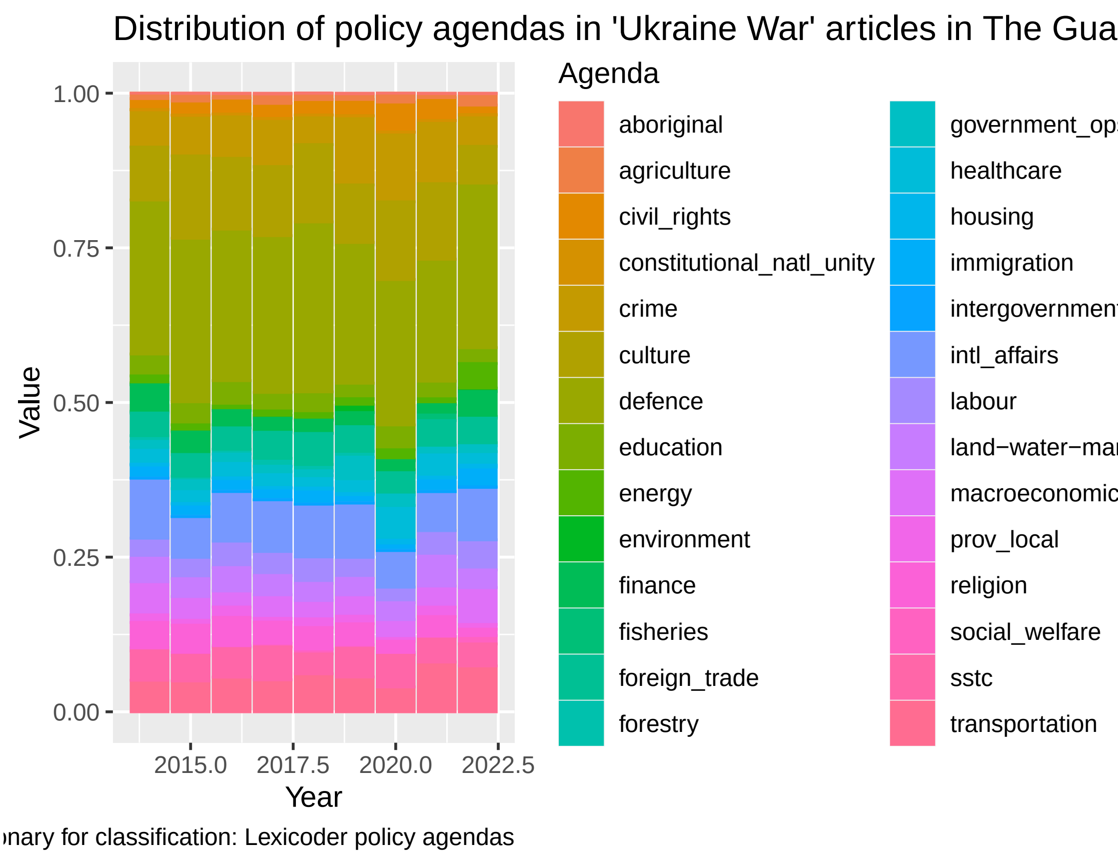


Sentiment Analysis

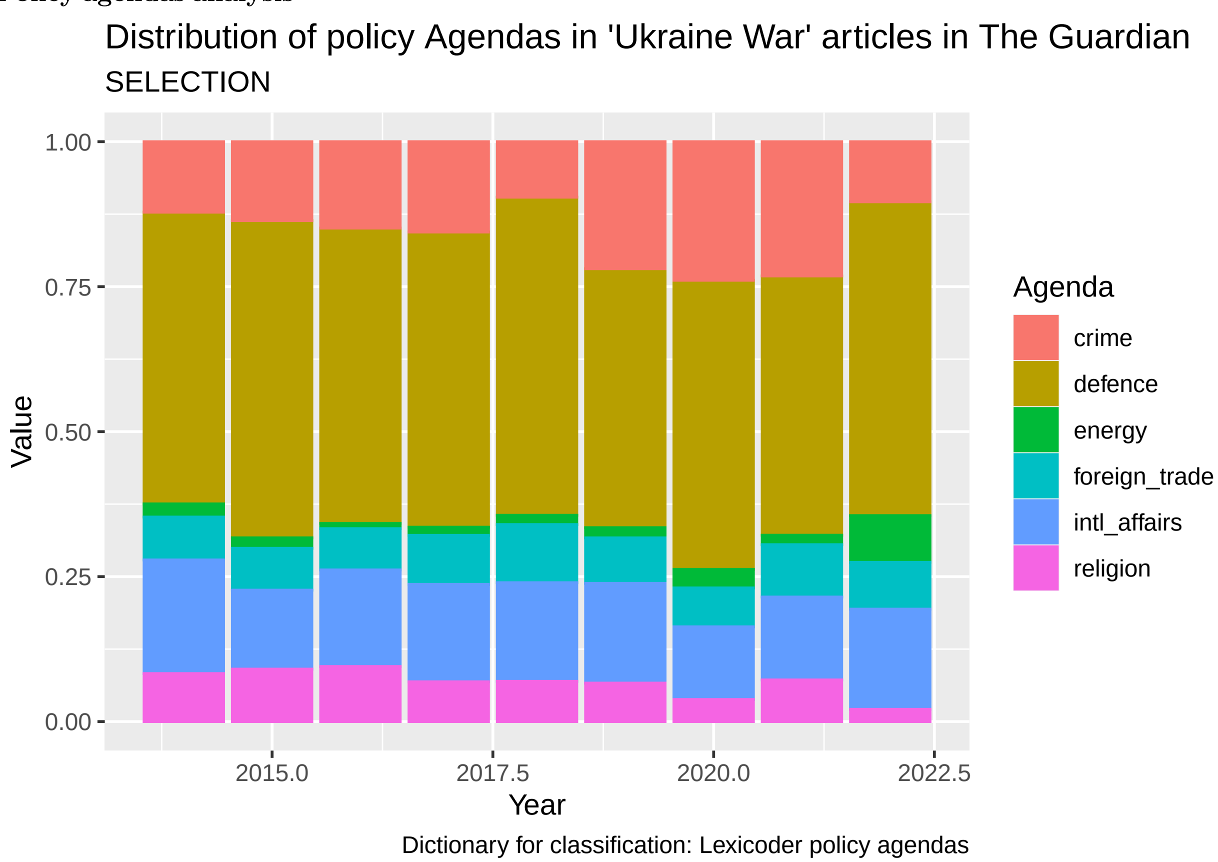


Word Frequency

```
##           word frequency
## 1          war      6424
## 2        peopl      4819
## 3       ukrain      4593
## 4       russia      4298
## 5      russian      4141
## 6      countri      3395
## 7       govern      3256
## 8        trump      2949
## 9         last      2723
## 10      presid      2578
## 11      report      2556
## 12       state      2511
## 13        forc      2395
## 14        mani      2345
## 15        week      2271
## 16      includ      2241
## 17      nation      2207
## 18       polit      2148
## 19          uk      2138
## 20       putin      2121
## 21    militari      2079
## 22   ukrainian      2003
## 23     support      1969
## 24        sinc      1732
## 25       still      1706
## 26       group      1674
## 27      minist      1657
## 28      intern      1655
## 29        citi      1646
## 30        hous      1596
## 31       month      1583
## 32       secur      1578
## 33        told      1577
## 34       power      1571
## 35      public      1560
## 36      former      1498
## 37      leader      1478
## 38     british      1425
## 39      famili      1420
## 40      offici      1420
## 41       parti      1395
## 42      attack      1375
## 43       chang      1373
## 44     foreign      1371
## 45       anoth      1328
## 46       start      1320
## 47        kill      1312
## 48       elect      1305
## 49        talk      1298
## 50          eu      1293
```



Policy agendas analysis



New York Times

URL generator for the New York Times API

```
nyt_key = read_lines("nyt_key.txt")

nyt_url <- function(search_word, date_from='', date_to='') {
  search_word <- str_replace(search_word, ' ', '%20')

  if (date_from == '' | date_to == '') {
    url <- paste0('http://api.nytimes.com/svc/search/v2/articlesearch.json?q=', search_word,
                  '&api-key=', nyt_key, sep='')
  } else {
    url <- paste0('http://api.nytimes.com/svc/search/v2/articlesearch.json?q=', search_word,
                  '&begin_date=', date_from, '&end_date=', date_to,
                  '&api-key=', nyt_key, sep='')
  }
  url

}
```

Afghanistan War

```

nyt_afgh_list <- vector("list")

counter <- 1

# in the NYT API it is only possible to search until page 200

for (year in 1999:2022) {
  Sys.sleep(6)
  base_url <- nyt_url('afghanistan war', paste0(year, '-01-01'),
                      paste0(year, '-12-31'))
  print(base_url)
  n_results <- fromJSON(base_url) %>% .$response %>% .$meta %>% .$hits
  max_pages <-  ceiling(((n_results / 10)-1))

  if (max_pages > 200) {
    max_pages <- 200
  } else {}

  print(year)

  for(i in 1:(max_pages/10)){
      tryCatch({
      print(i)
      url <- paste0(base_url, "&page=", i, sep='')
        NYTSearch <- fromJSON(url, flatten = TRUE) %>%
          data.frame(.,stringsAsFactors = FALSE)
      nyt_afgh_list[[counter]] <- NYTSearch
      counter <- counter + 1
      Sys.sleep(6)
      }, error=function(e){
        message(paste0("Error at ", year, " Page:", i))
```

```
    })
  }

}

nyt_afgh_results <- rbind_pages(nyt_afgh_list)

save(nyt_afgh_results, file = "Data/nyt_afgh_results.RData")
```

Get Data

```
load("Data/nyt_afgh_results.RData")

glimpse(nyt_afgh_results)

# basic data cleaning and selection of important variables

nyt_afgh <- nyt_afgh_results %>%
  filter(response.docs.type_of_material == "News")%>%
  select(abstract = response.docs.abstract,
         date = response.docs.pub_date,
         keywords = response.docs.keywords,
         url = response.docs.web_url,
         word_count = response.docs.word_count,
         headline = response.docs.headline.main,
         id = response.docs._id,
         hits_year = response.meta.hits )

summary(nyt_afgh$word_count)

nyt_afgh <- nyt_afgh %>%
  filter(word_count >=50)

nyt_afgh$date <- nyt_afgh$date %>%
  gsub("T.*",
       "", .) %>%
  as.Date

nyt_afgh$year <- nyt_afgh$date %>%
  gsub("([0-9]{4}).*",
       "\\1", .) %>% as.numeric

glimpse(nyt_afgh)

save(nyt_afgh, file = "Data/nyt_afgh.RData")
```

Data Cleaning

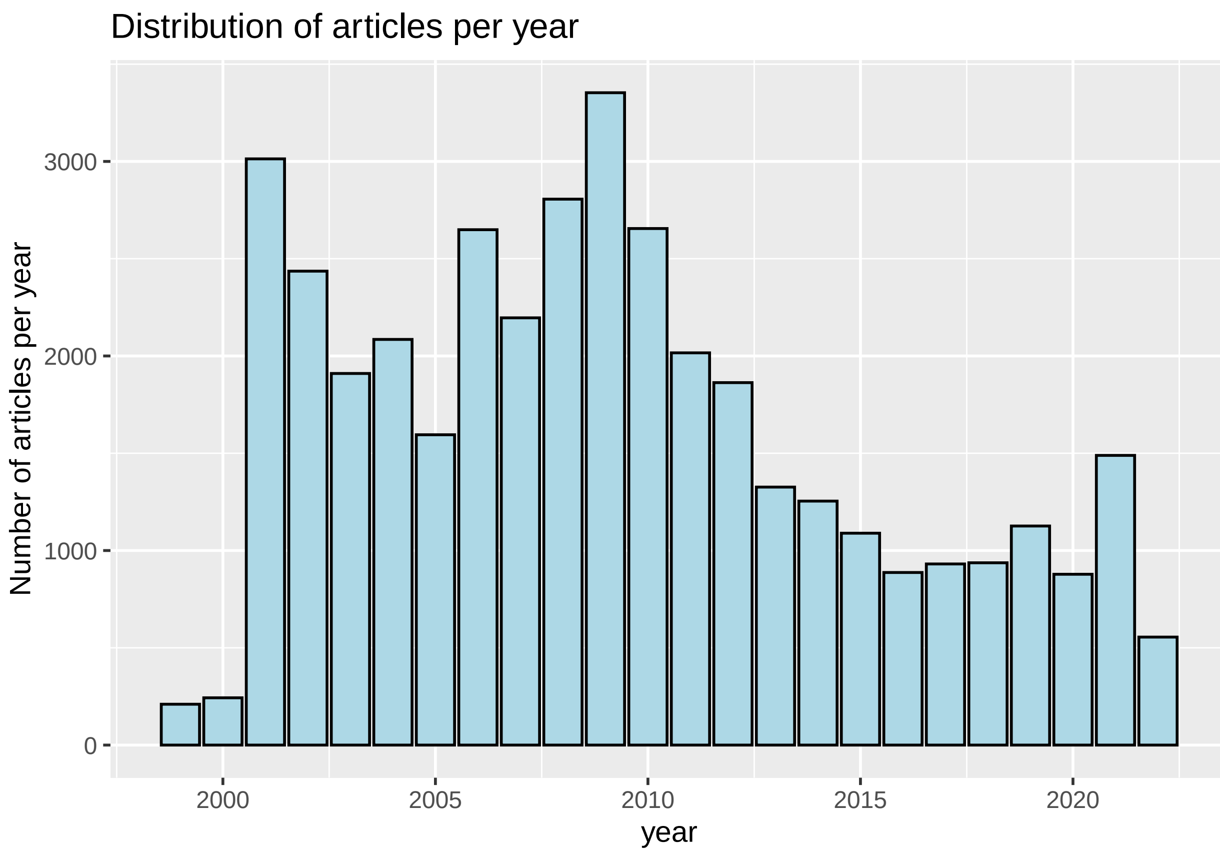
```
load("Data/nyt_afgh.RData")

# the NYT API does not automatically include the article text. That's why I had to write a script to au
```

```r
# Some sites are no longer available
# That's why I need to remove the article at position 978

nyt_afgh <- nyt_afgh[-c(978),]

article_text <- vector("list")
count <- 1
error_tries <- 0


while(count <= length(nyt_afgh$url)) {
  tryCatch({
  url = nyt_afgh$url[count]
  raw <- read_html(url)
  text <- html_nodes(raw, "section p") %>% html_text() %>% paste(., collapse=" ")
  article_text[[count]] <- text
  print(count)
  count <- count + 1
  }, error=function(e){
        message(paste0("Error at ", count))
        if (error_tries == 0) {
          error_tries <- error_tries + 1
          Sys.sleep(30)
        } else {
          article_text[[count]] <- NA
          error_tries <- 0
          count <- count + 1
        }
      })
}




nyt_afgh$text <- article_text %>% as.character

glimpse(nyt_afgh)
head(nyt_afgh)

nyt_afgh <- nyt_afgh %>% filter(!is.na(nyt_afgh$text))

save(nyt_afgh, file = "Data/nyt_afgh.RData")
```

Get Full articles

```r
load("Data/nyt_afgh.RData")

# Create a histogram to show the distribution of articles

nyt_afgh_hist <- ggplot(nyt_afgh, aes(x=date))+
  geom_histogram(fill="lightblue", color="black", binwidth = 200)
# nyt_afgh_hist
```

```
nyt_afgh_hist_year <- nyt_afgh[!duplicated(nyt_afgh[ , c("hits_year")]), ] %>%
  select(hits_year, year) %>%
  ggplot(aes(x=year, y=hits_year))+
  geom_bar(stat='identity', fill = "lightblue", color="black")+
  labs(title = "Distribution of articles per year", y="Number of articles per year")
nyt_afgh_hist_year
```



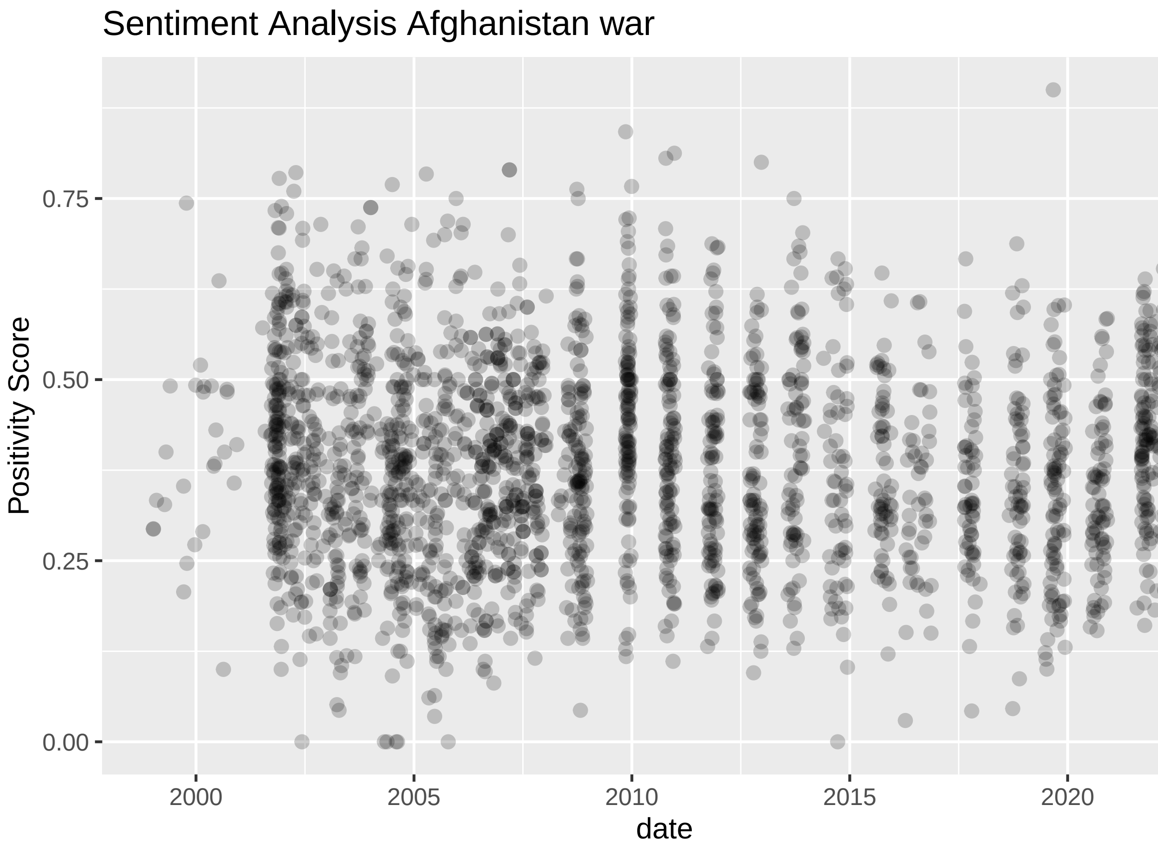
Basic Analysis

```
load("Data/nyt_afgh.RData")

nyt_afgh_corpus <- corpus(nyt_afgh,
                          text_field = "text")

summary(nyt_afgh_corpus) %>% head
docvars(nyt_afgh_corpus) %>%
  head

nyt_afgh_corpus[1]


nyt_afgh_toks <- tokens(nyt_afgh_corpus,
                        what = c("word"),
                        remove_separators = TRUE,
                        include_docvars = TRUE,
                        ngrams = 1L,
```

```r
                    remove_numbers = FALSE,
                    remove_punct = TRUE,
                    remove_symbols = FALSE,
                    remove_hyphens = FALSE)

# nyt_afgh_toks %>% head

nyt_afgh_toks <- nyt_afgh_toks %>%
  tokens_tolower %>%
  tokens_remove(stopwords("english"), padding = TRUE) %>%
  tokens_remove("") %>%
  tokens_wordstem(language = "english")

save(nyt_afgh_toks, file = "Data/nyt_afgh_toks.RData")
```

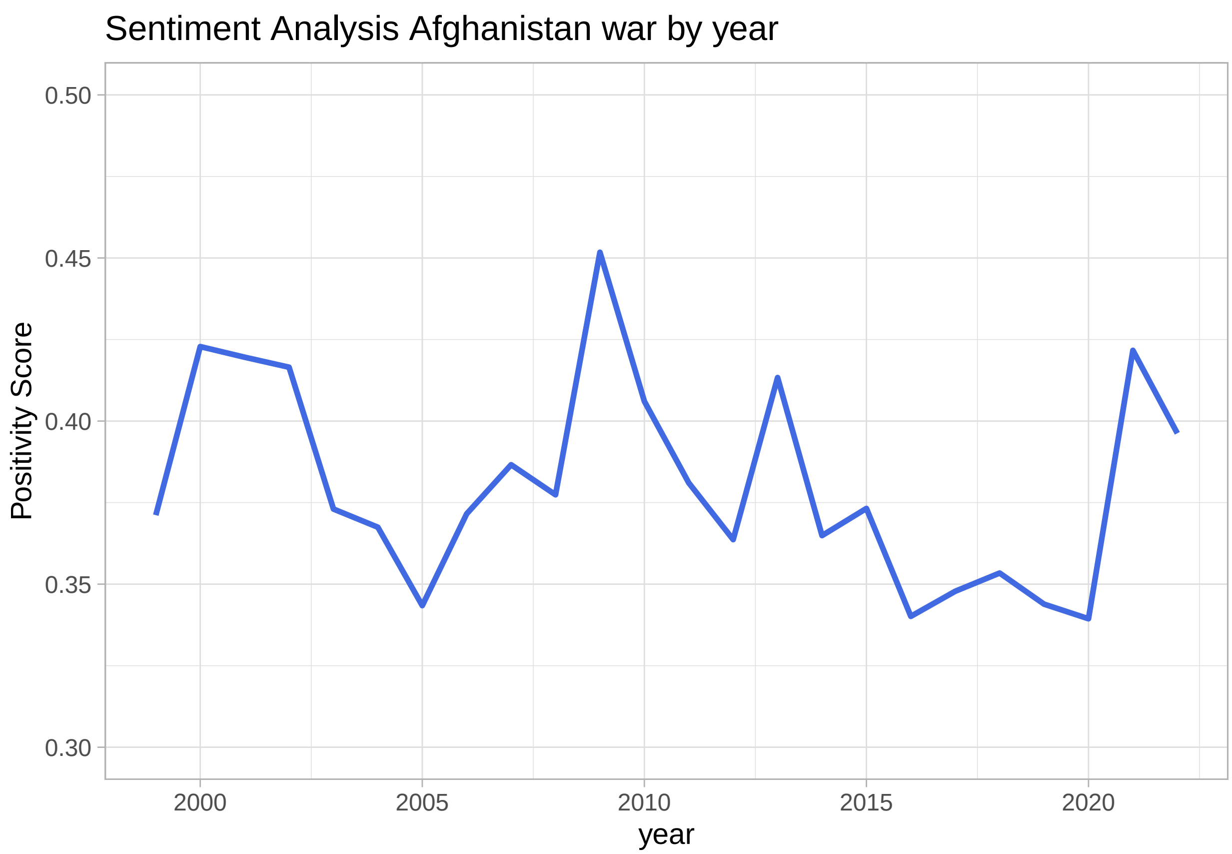
Create a corpus and tokens

```r
load("Data/nyt_afgh_toks.RData")
load("Data/nyt_afgh.RData")

# make a sentiment analysis on the ratio of positive to negative words in each article

nyt_afgh_toks_sent <- tokens_lookup(nyt_afgh_toks,
                                    dictionary =  data_dictionary_LSD2015[1:2])
# nyt_afgh_toks_sent %>% head

nyt_afgh_dfm_sent <- dfm(nyt_afgh_toks_sent)
# nyt_afgh_dfm_sent %>% head

nyt_afgh_pos_neg <- nyt_afgh_dfm_sent %>% convert(.,to = "data.frame") %>%
  mutate(pos_to_neg = positive / (positive + negative))

# summary(nyt_afgh_pos_neg)
#nyt_afgh_pos_neg %>% filter(is.na(nyt_afgh_pos_neg$pos_to_neg))
# 1 NA in year 2002
# Row has 0 negative and 0 positive -> NaN -> discard

nyt_afgh_sent <- nyt_afgh

nyt_afgh_sent$pos_neg <- nyt_afgh_pos_neg$pos_to_neg
nyt_afgh_sent <- nyt_afgh_sent %>% filter(!is.na(nyt_afgh_sent$pos_neg))

plot_nyt_afgh_sent <- ggplot(nyt_afgh_sent, aes(x=date, y=pos_neg))+
  geom_point(size=2, alpha=0.2)+
  labs(title = "Sentiment Analysis Afghanistan war", y="Positivity Score")
plot_nyt_afgh_sent
```



Sentiment Analysis

```
nyt_afgh_by_year <- nyt_afgh_sent$pos_neg %>%
  aggregate(by=list(nyt_afgh_sent$year), FUN = mean) %>% rename(year = Group.1,
                                                                 pos_neg = x)
# nyt_afgh_by_year

plot_nyt_afgh_pos_by_year <- ggplot(nyt_afgh_by_year, aes(x=year, y=pos_neg))+
  geom_line(color="royalblue", size=1, aes(group=1))+
  theme_light()+
  ylim(0.3, 0.5)+
  labs(title = "Sentiment Analysis Afghanistan war by year", y="Positivity Score")
plot_nyt_afgh_pos_by_year
```



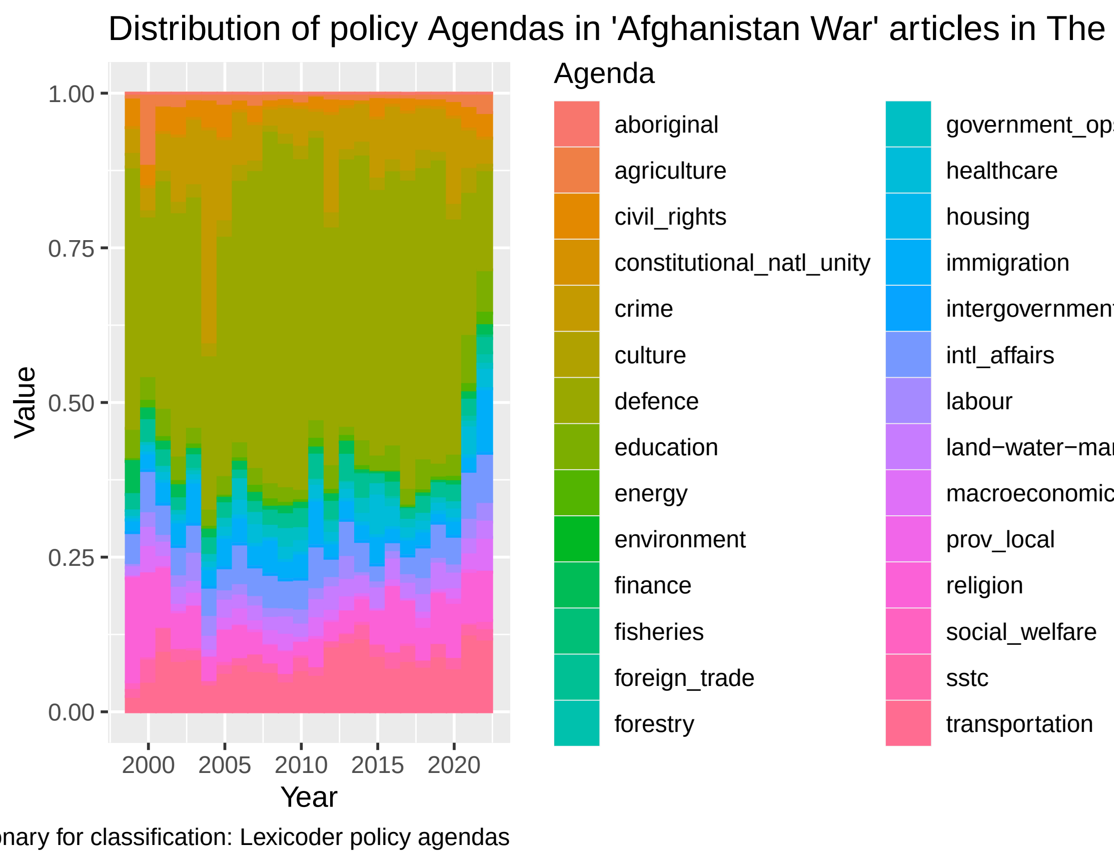

```
load("Data/nyt_afgh_toks.RData")


# load list with the most common words in English to remove them from the tokens

common_words <- read.delim("Data/1-1000.txt", header = FALSE) %>%
  head(200) %>%
  as.vector()
common_words <- common_words$V1

# removing the top 200 words from the tokens
nyt_afgh_toks_wordcount <- nyt_afgh_toks %>%
  tokens_remove(common_words)

nyt_afgh_dfm <- dfm(nyt_afgh_toks_wordcount)



nyt_afgh_topwords <- topfeatures(nyt_afgh_dfm, 50) %>%
  data.frame(word=names(.),
             frequency =.,
             row.names = c())

nyt_afgh_topwords
```

Word Frequency

```
##             word frequency
## 1   afghanistan     15253
## 2            mr     14062
## 3       taliban     13462
## 4        afghan     12379
## 5      american     11601
## 6          forc      9176
## 7        offici      9136
## 8          kill      8058
## 9      militari      7440
## 10         unit      7108
## 11       govern      6710
## 12        state      6550
## 13       attack      5754
## 14          war      5511
## 15        secur      5239
## 16        troop      5207
## 17      countri      5110
## 18        peopl      4605
## 19        polic      4415
## 20        offic      4396
## 21      soldier      4323
## 22        kabul      4235
## 23      provinc      4152
## 24       presid      4141
## 25       nation      3862
## 26      command      3860
## 27       report      3721
## 28     pakistan      3459
## 29     civilian      3312
## 30         last      3276
## 31         mani      3176
## 32      general      3145
## 33         oper      3120
## 34       karzai      2918
## 35        month      2883
## 36       member      2864
## 37        fight      2843
## 38         nato      2841
## 39       includ      2820
## 40     district      2789
## 41        group      2788
## 42         area      2620
## 43         base      2599
## 44         week      2567
## 45         iraq      2560
## 46       insurg      2495
## 47         armi      2478
## 48      support      2403
## 49         bomb      2357
## 50       leader      2321
```

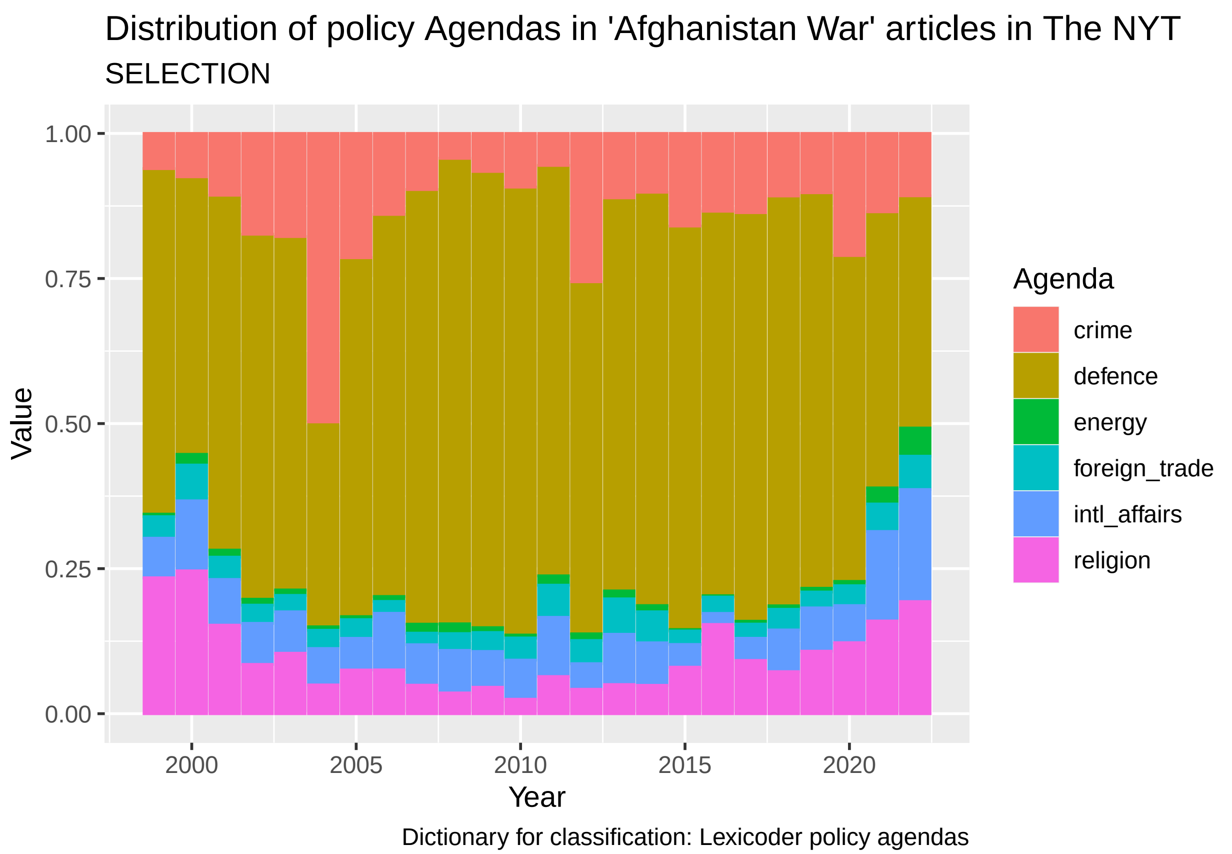
```r
save(nyt_afgh_dfm, file = "Data/nyt_afgh_dfm.RData")
```

```r
load("Data/nyt_afgh_toks.RData")
load("Data/nyt_afgh.RData")

# Load the Lexicoder policy agendas
policyagendas <- dictionary(file = "Data/policy_agendas_english.lcd")

# lookup the policy agendas dictionary and give each article a score
nyt_afgh_toks_pol <- tokens_lookup(nyt_afgh_toks, dictionary = policyagendas)
nyt_afgh_pol <- dfm(nyt_afgh_toks_pol) %>%
  convert(to = "data.frame") %>%
  select(-doc_id)

# divide the values for each row through the sum of each row to get relative values for the agendas for
nyt_afgh_pol <- nyt_afgh_pol / rowSums(nyt_afgh_pol)

nyt_afgh_pol$year <- nyt_afgh$year

nyt_afgh_pol <- drop_na(nyt_afgh_pol)

nyt_afgh_pol_by_year <- nyt_afgh_pol %>%
  group_by(year) %>%
  summarise_each(funs = sum)

# Plot the results to better inspect them
nyt_afgh_pol_by_year_plot1 <- nyt_afgh_pol_by_year %>%
  pivot_longer(cols = 2:29, names_to = "Agenda") %>%
  ggplot(aes(x=year, y=value, colour = Agenda, fill = Agenda))+
  geom_bar(position="fill", stat="identity")+
  labs(x="Year", y="Value", title="Distribution of policy Agendas in 'Afghanistan War' articles in The N
       caption = "Dictionary for classification: Lexicoder policy agendas")
nyt_afgh_pol_by_year_plot1
```



Policy agendas analysis

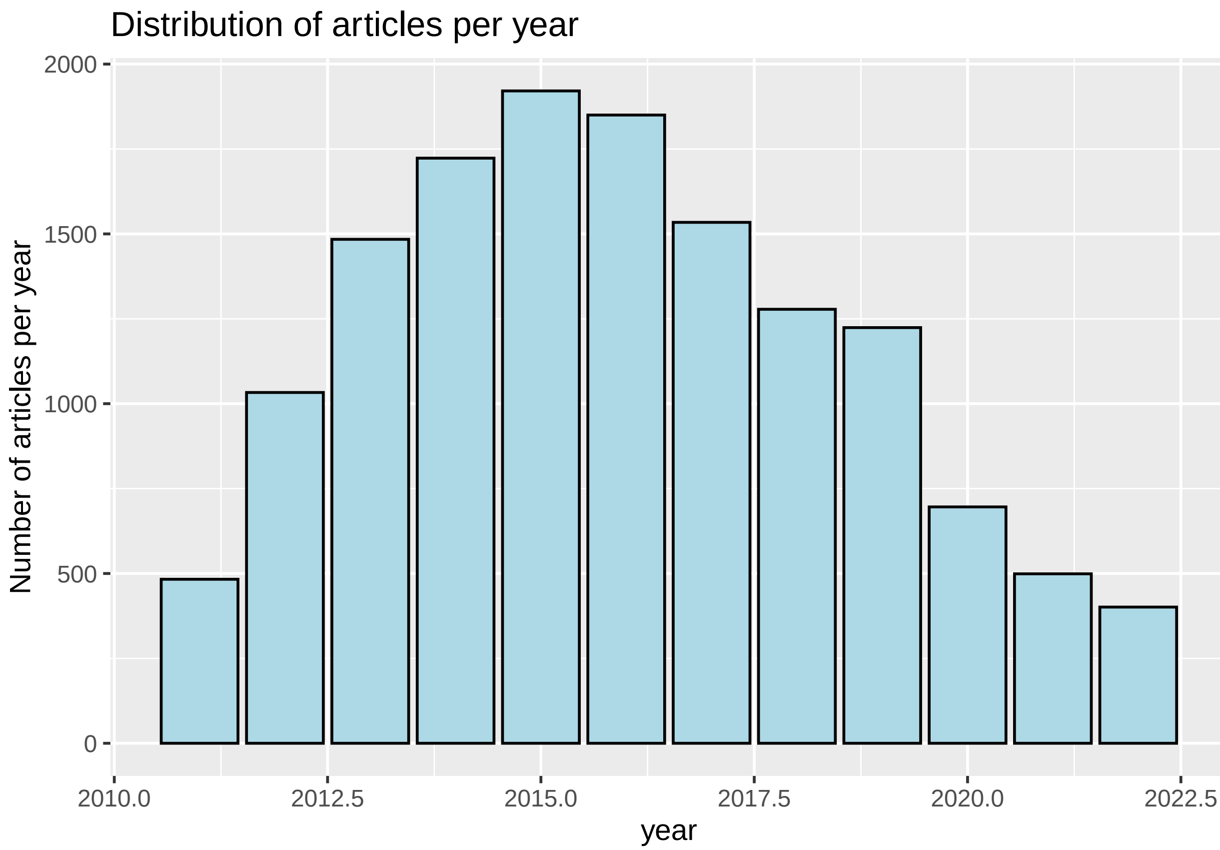
```
# select some agendas which seam important
nyt_afgh_pol_by_year_plot_select <- nyt_afgh_pol_by_year %>%
  select(year, defence, energy, foreign_trade, intl_affairs, religion, crime) %>%
  pivot_longer(cols = 2:7, names_to = "Agenda") %>%
  ggplot(aes(x=year, y=value, colour = Agenda, fill = Agenda))+
  geom_bar(position="fill", stat="identity")+
  labs(x="Year", y="Value", title="Distribution of policy Agendas in 'Afghanistan War' articles in The N
       subtitle = "SELECTION", caption = "Dictionary for classification: Lexicoder policy agendas")
nyt_afgh_pol_by_year_plot_select
```



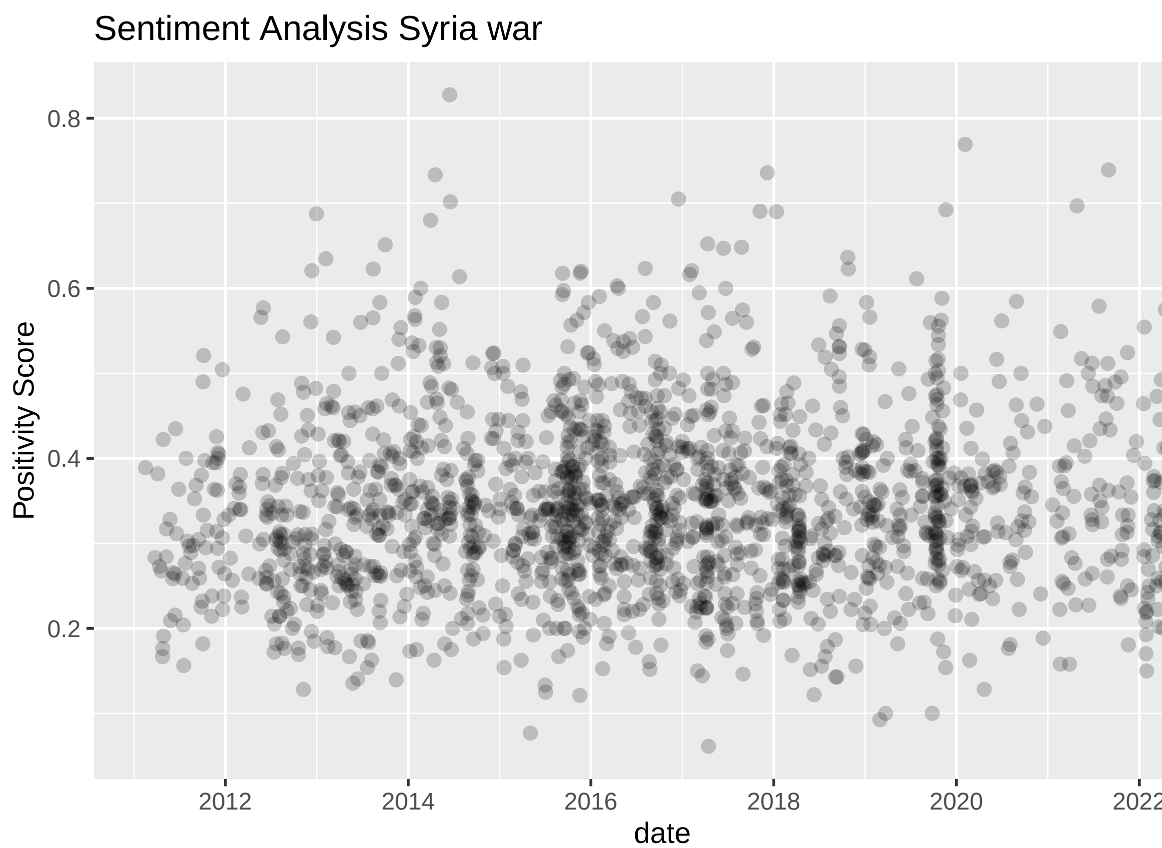

```
# nyt_afgh_pol_by_year

save(nyt_afgh_pol, file = "Data/nyt_afgh_pol.RData")
```

The code for the other wars are not included in the rendered markdown since they are almost identical to the coverage of the Afghanistan War

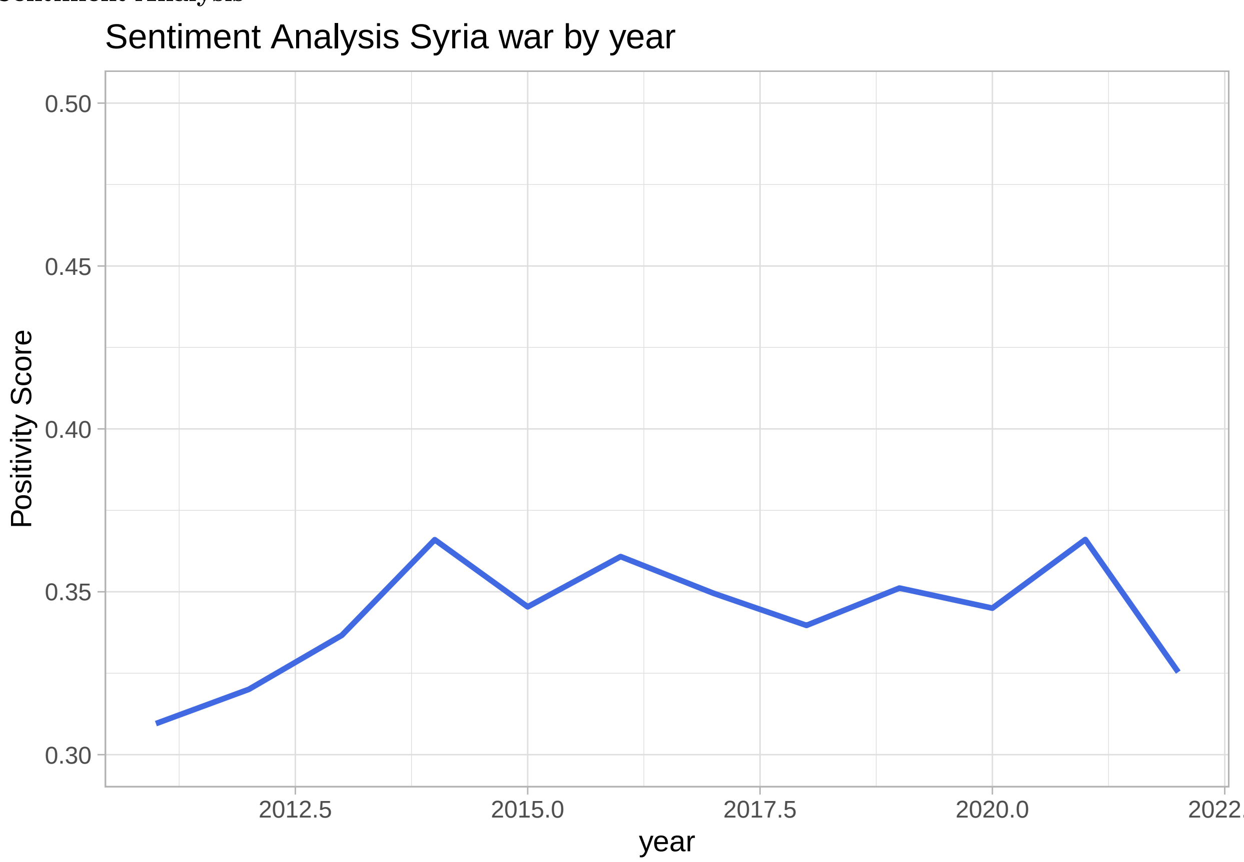
Syria War



Basic Analysis

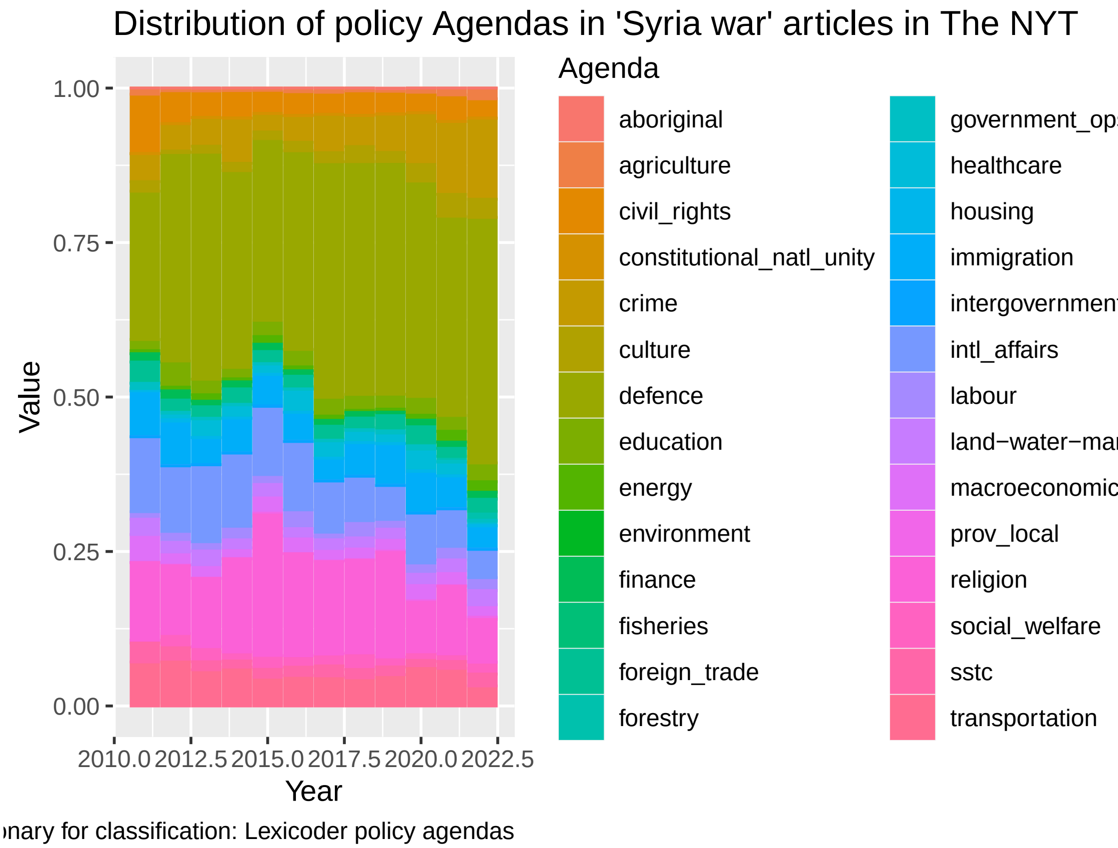


Sentiment Analysis

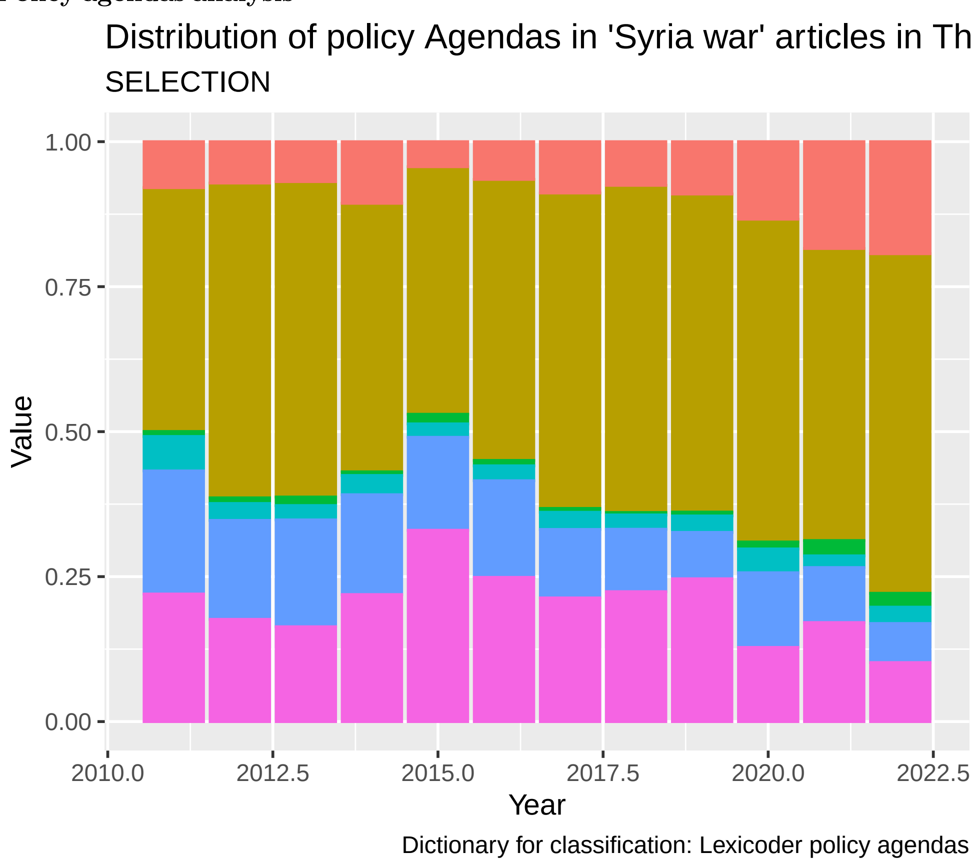


Word Frequency

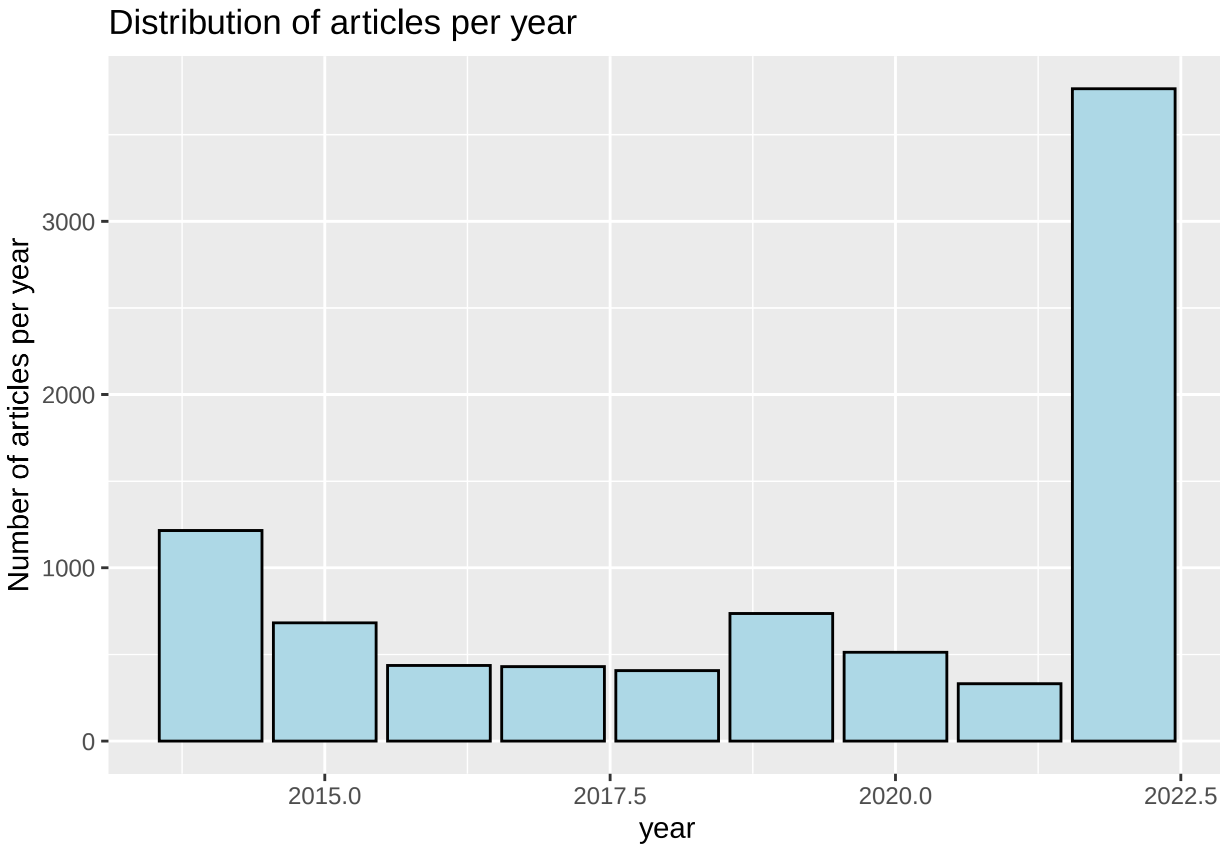
```
##         word frequency
## 1      syria     16362
## 2         mr     16331
## 3     syrian     13383
## 4      state     12716
## 5     govern      9394
## 6        war      9154
## 7       unit      8299
## 8       forc      8088
## 9      group      7090
## 10  american      5468
## 11     islam      5426
## 12     peopl      5189
## 13  militari      5046
## 14    offici      4954
## 15    presid      4874
## 16   countri      4864
## 17    russia      4794
## 18     rebel      4755
## 19    turkey      4461
## 20  conflict      4403
## 21    attack      4335
## 22  al-assad      4325
## 23      kill      4134
## 24    nation      3984
## 25     fight      3933
## 26     assad      3795
## 27       arm      3745
## 28    report      3579
## 29   russian      3554
## 30      mani      3514
## 31      iraq      3360
## 32   support      3347
## 33   kurdish      3265
## 34   fighter      3231
## 35    weapon      3228
## 36      took      2977
## 37     secur      2899
## 38      iran      2859
## 39      area      2801
## 40      last      2796
## 41 territori      2689
## 42    includ      2663
## 43      bomb      2644
## 44     trump      2602
## 45     began      2551
## 46      citi      2539
## 47    refuge      2522
## 48    chemic      2484
## 49   control      2474
## 50   foreign      2466
```



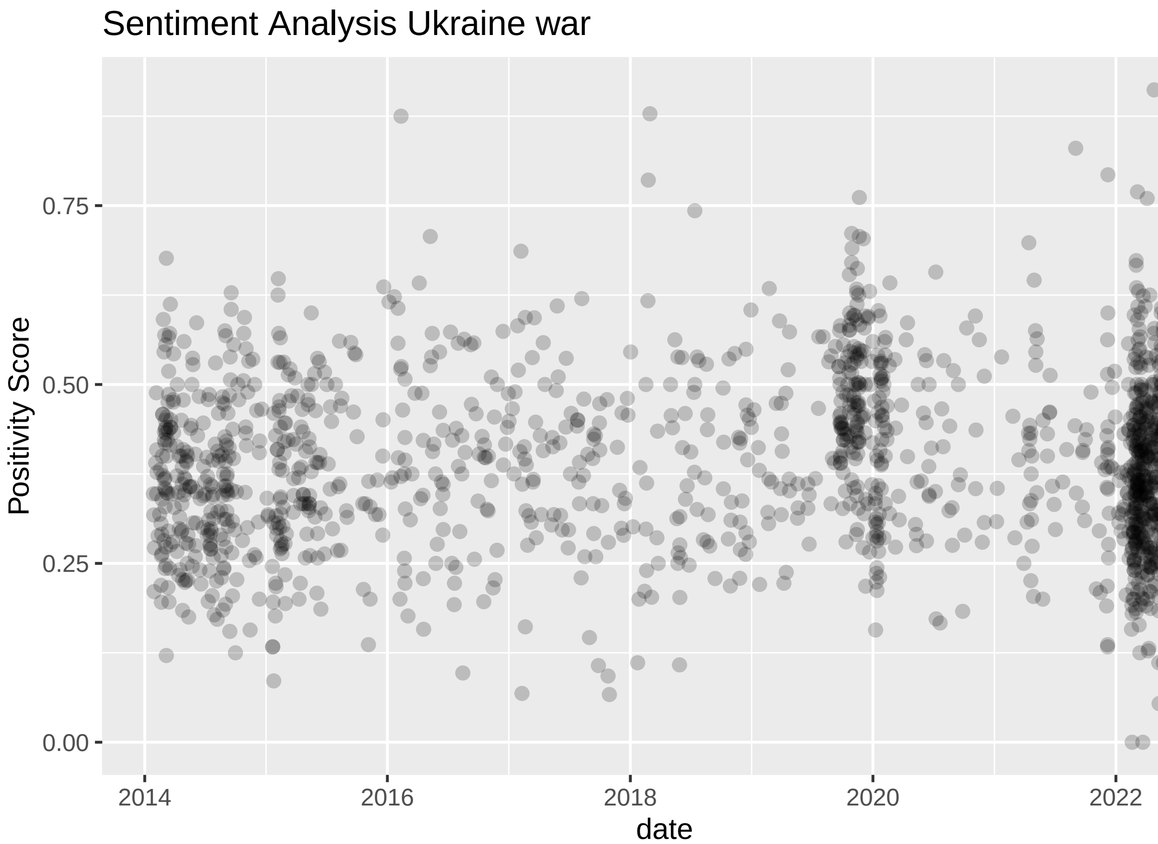
Policy agendas analysis



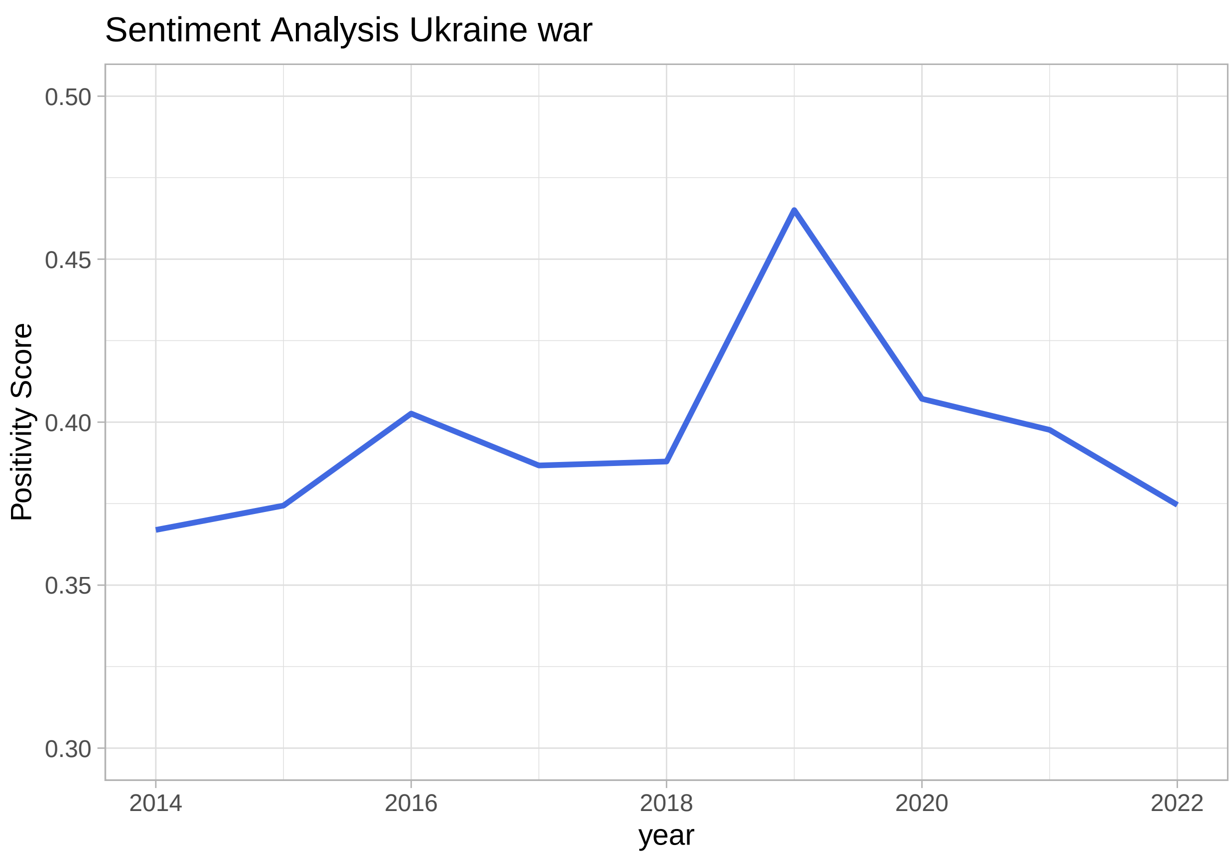
Ukraine War



Basic Analysis

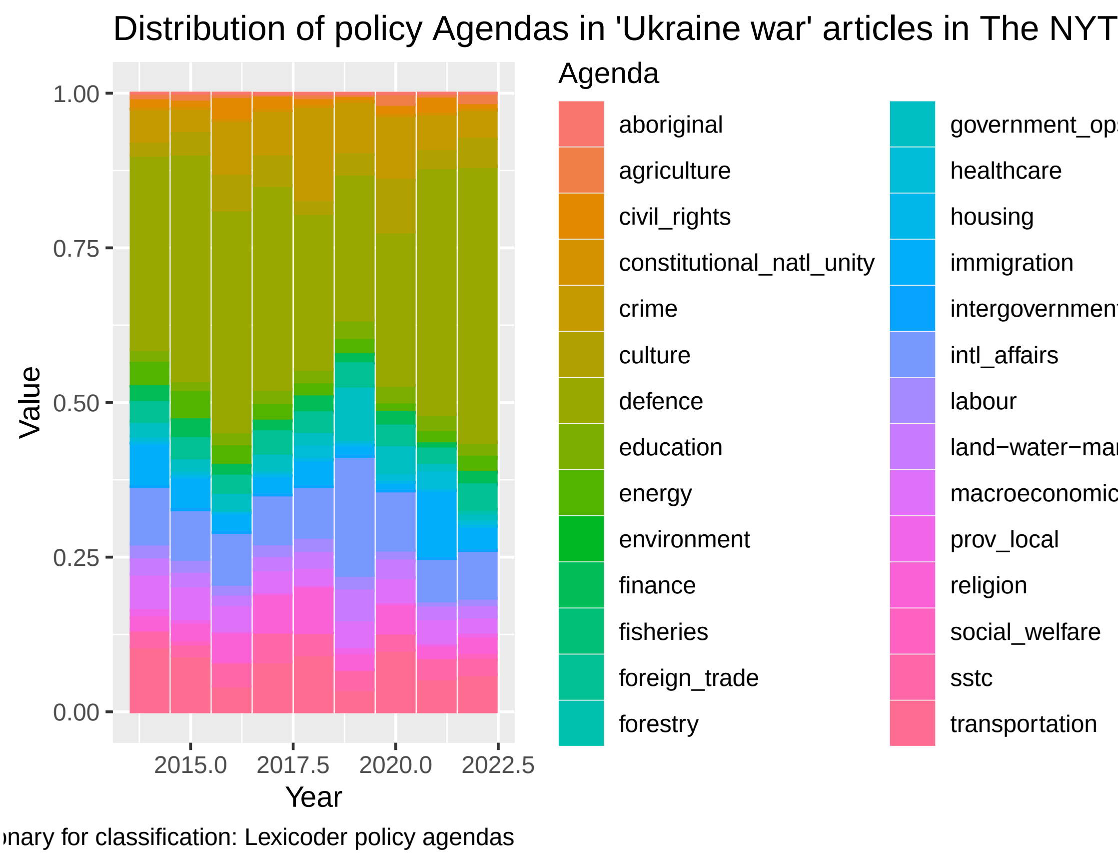


Sentiment Analysis

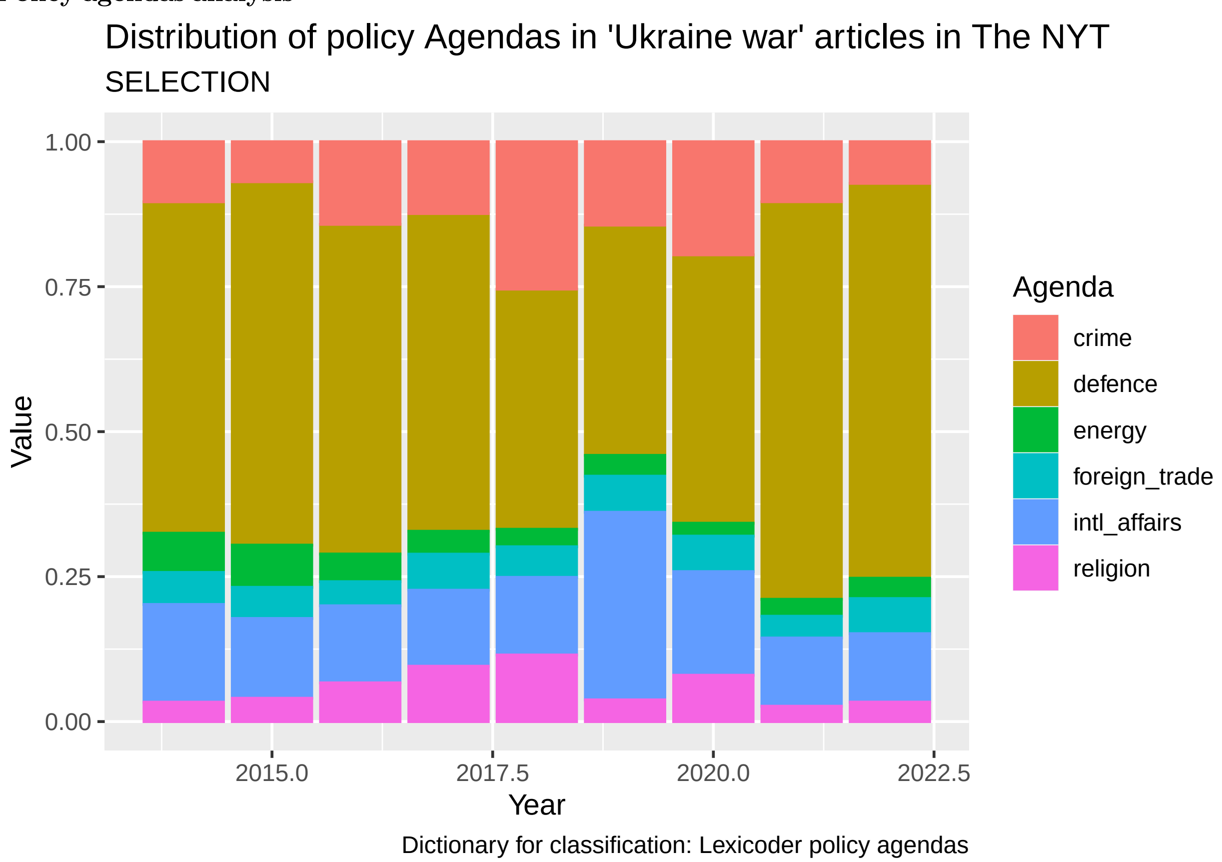


Word Frequency

```
##         word frequency
## 1         mr     17994
## 2     ukrain     14357
## 3     russia     10526
## 4    russian     10172
## 5  ukrainian      7869
## 6     presid      6534
## 7        war      6147
## 8      state      5141
## 9   militari      4508
## 10     putin      4301
## 11     trump      4297
## 12    offici      4247
## 13   countri      4166
## 14      unit      3968
## 15      forc      3464
## 16     peopl      3429
## 17    govern      3388
## 18  american      2771
## 19    nation      2683
## 20    report      2402
## 21     polit      2401
## 22      last      2307
## 23    moscow      2278
## 24     secur      2270
## 25    includ      2246
## 26      hous      2223
## 27  european      2210
## 28      week      2164
## 29      mani      2109
## 30      citi      2101
## 31    leader      2098
## 32   support      2049
## 33  investig      1907
## 34   eastern      1897
## 35     fight      1846
## 36    former      1828
## 37        ms      1799
## 38     month      1740
## 39     europ      1714
## 40     troop      1705
## 41      sinc      1685
## 42       tri      1675
## 43    region      1670
## 44   foreign      1635
## 45     biden      1604
## 46    attack      1603
## 47     group      1582
## 48   soldier      1566
## 49     offic      1564
## 50  zelenski      1552
```



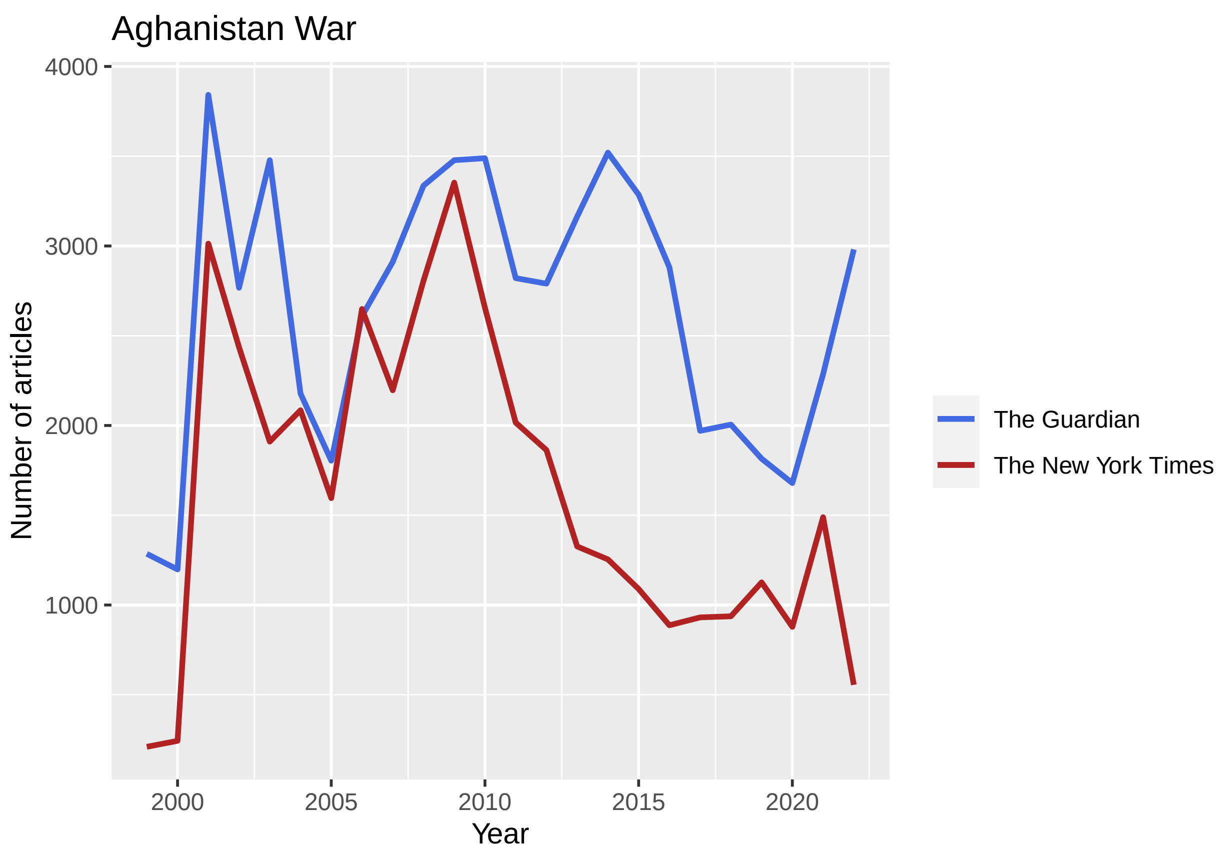
Policy agendas analysis



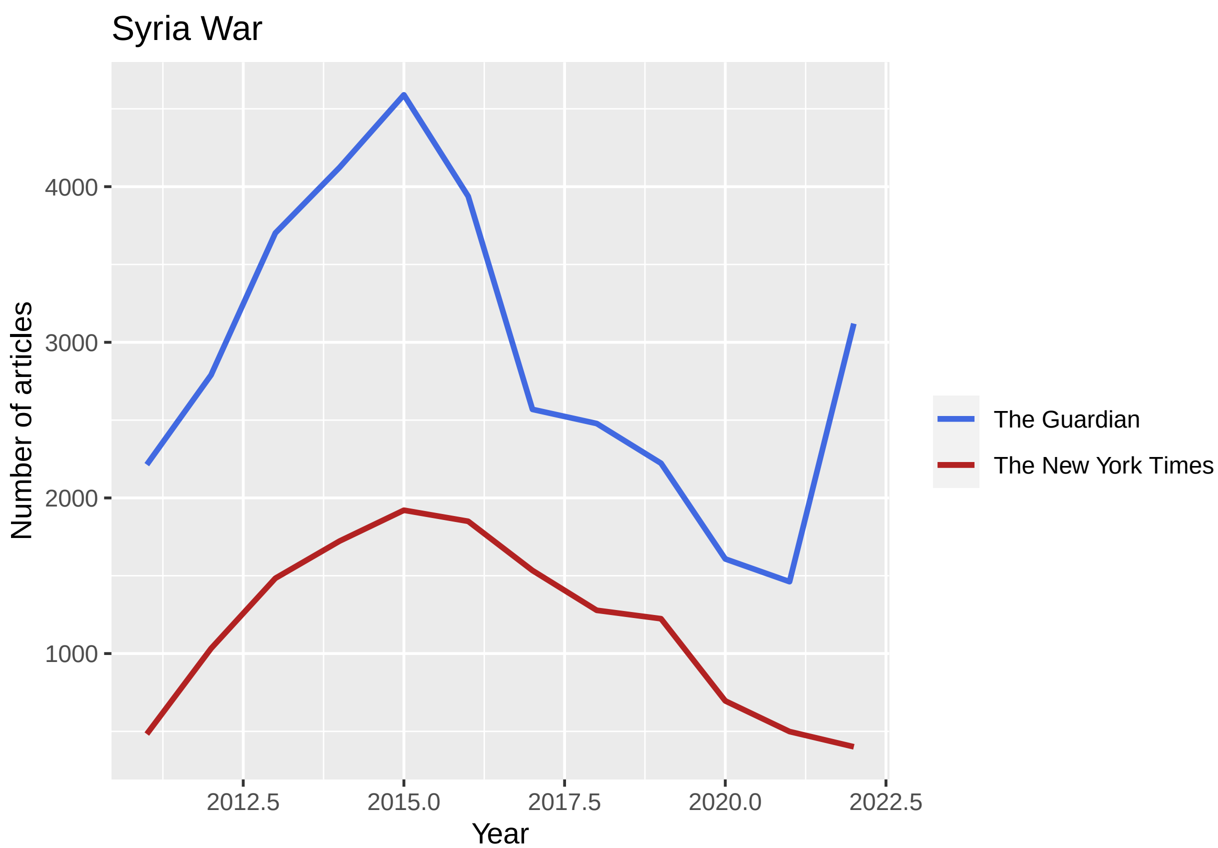
Comparison

Distribution

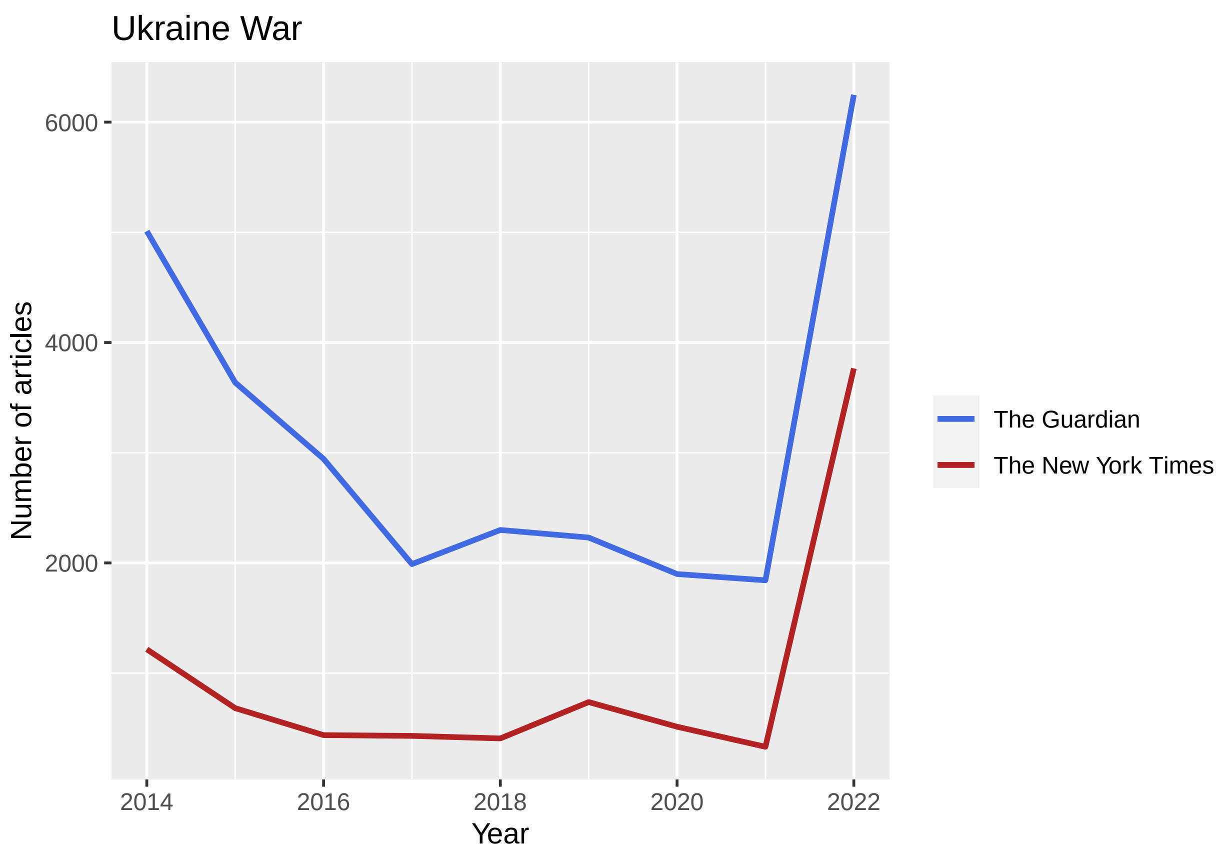
```
afgh_year_total <- afgh[!duplicated(afgh[ , c("year")]), ] %>%
  select(total_year, year)

syria_year_total <- syria[!duplicated(syria[ , c("year")]), ] %>%
  select(total_year, year)

ukraine_year_total <- ukraine[!duplicated(ukraine[ , c("year")]), ] %>%
  select(total_year, year)

nyt_afgh_year_total <- nyt_afgh[!duplicated(nyt_afgh[ , c("year")]), ] %>%
  select(hits_year, year)

nyt_syria_year_total <- nyt_syria[!duplicated(nyt_syria[ , c("year")]), ] %>%
  select(hits_year, year)

nyt_ukraine_year_total <- nyt_ukraine[!duplicated(nyt_ukraine[ , c("year")]), ] %>%
  select(hits_year, year)

afgh_dist_plot <- ggplot()+
  geom_line(afgh_year_total,
            mapping = aes(x=year, y=total_year,
                          color="The Guardian"), size=1)+
  geom_line(nyt_afgh_year_total,
            mapping = aes(x=year, y=hits_year,
                          color="The New York Times"), size=1)+
  scale_color_manual("",
                     values = c("The Guardian"="royalblue",
                                "The New York Times"="firebrick"))+
  labs(title="Aghanistan War", x="Year", y="Number of articles")
afgh_dist_plot
```




```
syria_dist_plot <- ggplot()+
  geom_line(syria_year_total,
            mapping = aes(x=year, y=total_year,
                          color="The Guardian"), size=1)+
  geom_line(nyt_syria_year_total,
            mapping = aes(x=year, y=hits_year,
                          color="The New York Times"), size=1)+
  scale_color_manual("",
                     values = c("The Guardian"="royalblue",
                                "The New York Times"="firebrick"))+
  labs(title="Syria War", x="Year", y="Number of articles")
syria_dist_plot
```




```
ukraine_dist_plot <- ggplot()+
  geom_line(ukraine_year_total,
            mapping = aes(x=year, y=total_year,
                          color="The Guardian"), size=1)+
  geom_line(nyt_ukraine_year_total,
            mapping = aes(x=year, y=hits_year,
                          color="The New York Times"), size=1)+
  scale_color_manual("",
                     values = c("The Guardian"="royalblue",
                                "The New York Times"="firebrick"))+
  labs(title="Ukraine War", x="Year", y="Number of articles")
ukraine_dist_plot
```

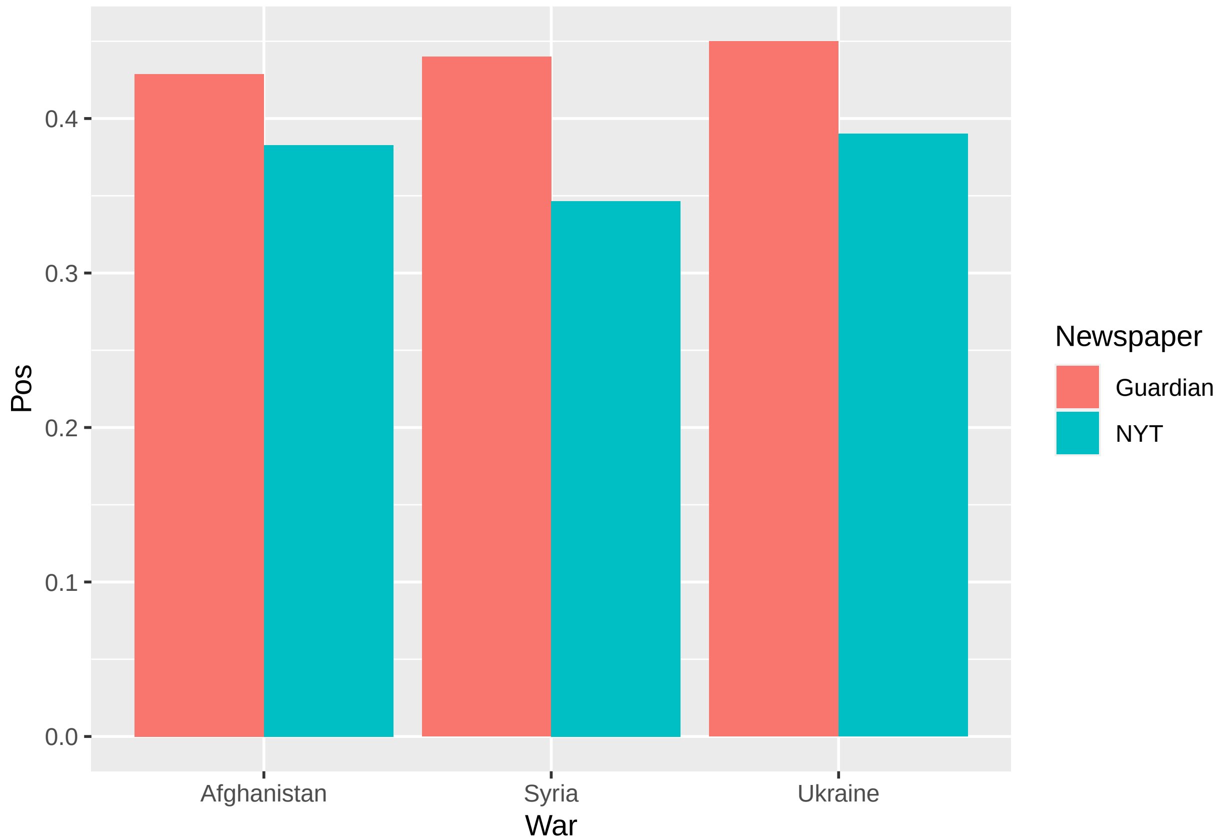


By looking at the distribution of the articles the first noticeable difference between the two newspaper is that the Guardian published more articles almost in every year about every war. But the distance between the two newspaper varies a lot. When comparing the war in Afghanistan and the war in Ukraine it is clearly visible that the difference in the number of published articles is a lot smaller in the Afghanistan war and a lot larger in the Ukraine war. This makes sense since the origin country of the New York Times was involved in the war and the origin country of the Guardian is geographically and politically much closer to the Ukraine. Another noticeable difference is that the number of articles published for the war in Syria peaks again in 2022 in the Guardian but there's no sign of that in the New York Times.

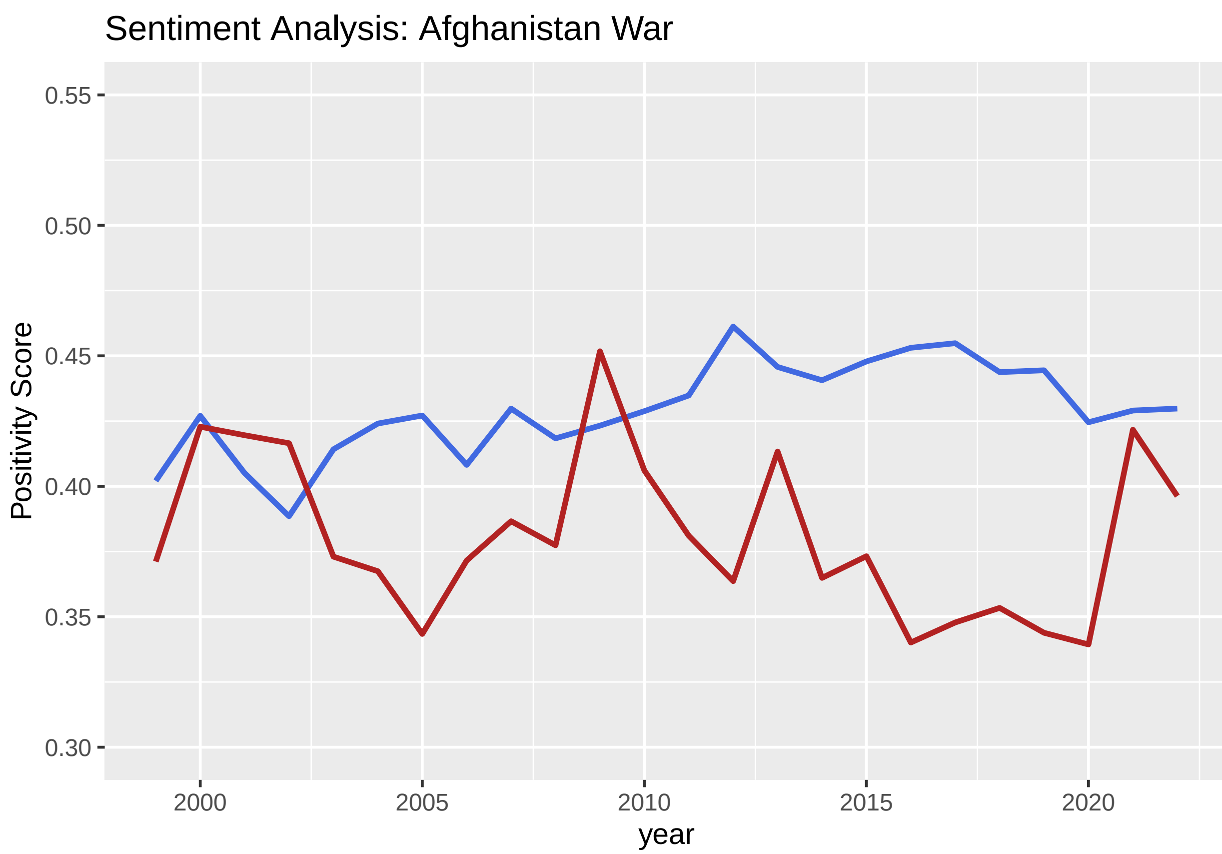
Sentiment Analysis

```
sent_comparison <- data.frame(matrix(nrow = 6, ncol = 3))
colnames(sent_comparison) <- c("Newspaper", "War", "Pos")
sent_comparison$Newspaper <- c("Guardian", "Guardian", "Guardian", "NYT", "NYT", "NYT")
sent_comparison$War <- c("Afghanistan", "Syria", "Ukraine", "Afghanistan", "Syria", "Ukraine")
sent_comparison$Pos <- c(mean(afgh_sent$pos_neg),
                         mean(syria_sent$pos_neg),
                         mean(ukraine_sent$pos_neg),
                         mean(nyt_afgh_sent$pos_neg),
                         mean(nyt_syria_sent$pos_neg),
                         mean(nyt_ukraine_sent$pos_neg))

# sent_comparison

comp_sent_plot <- sent_comparison %>%
  ggplot(aes(x=War, y=Pos, fill = Newspaper))+
```

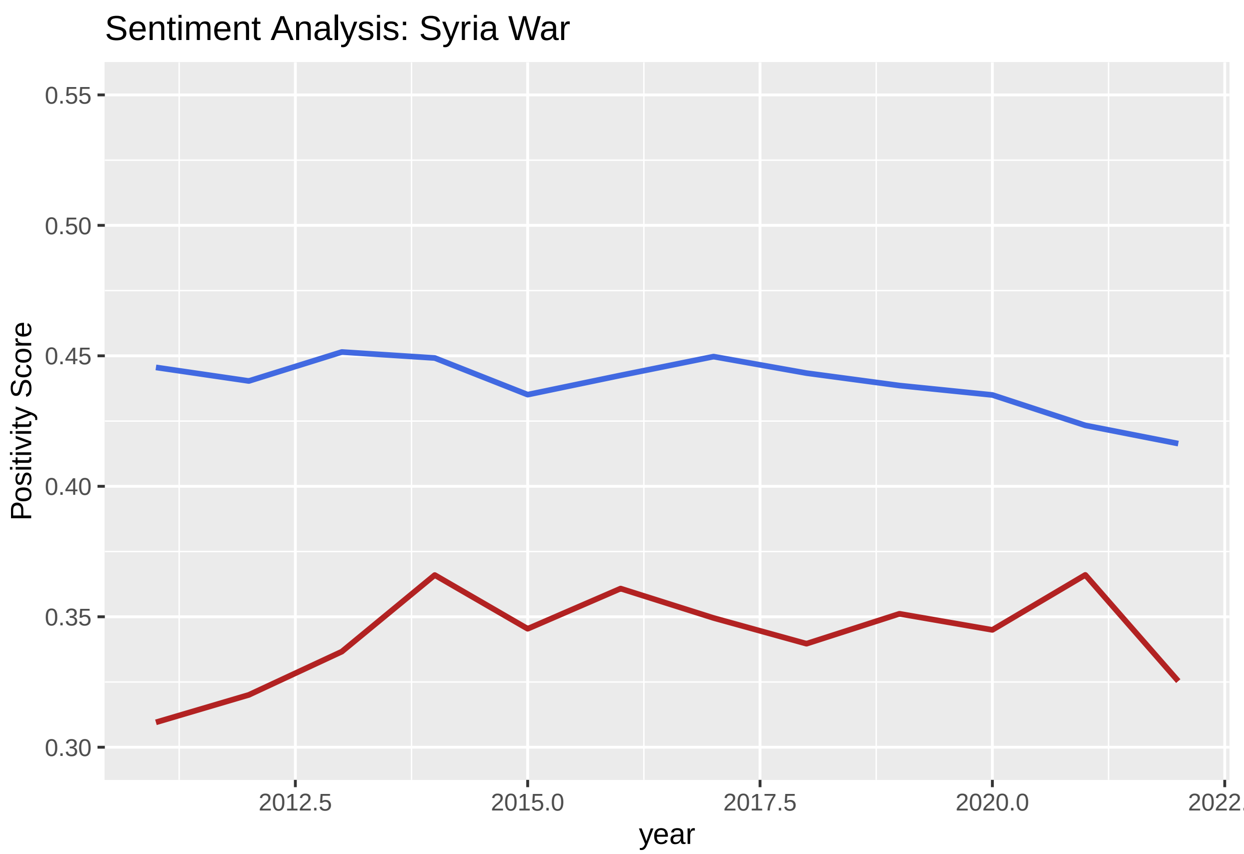
```
  geom_bar(position = "dodge", stat="identity")
comp_sent_plot
```



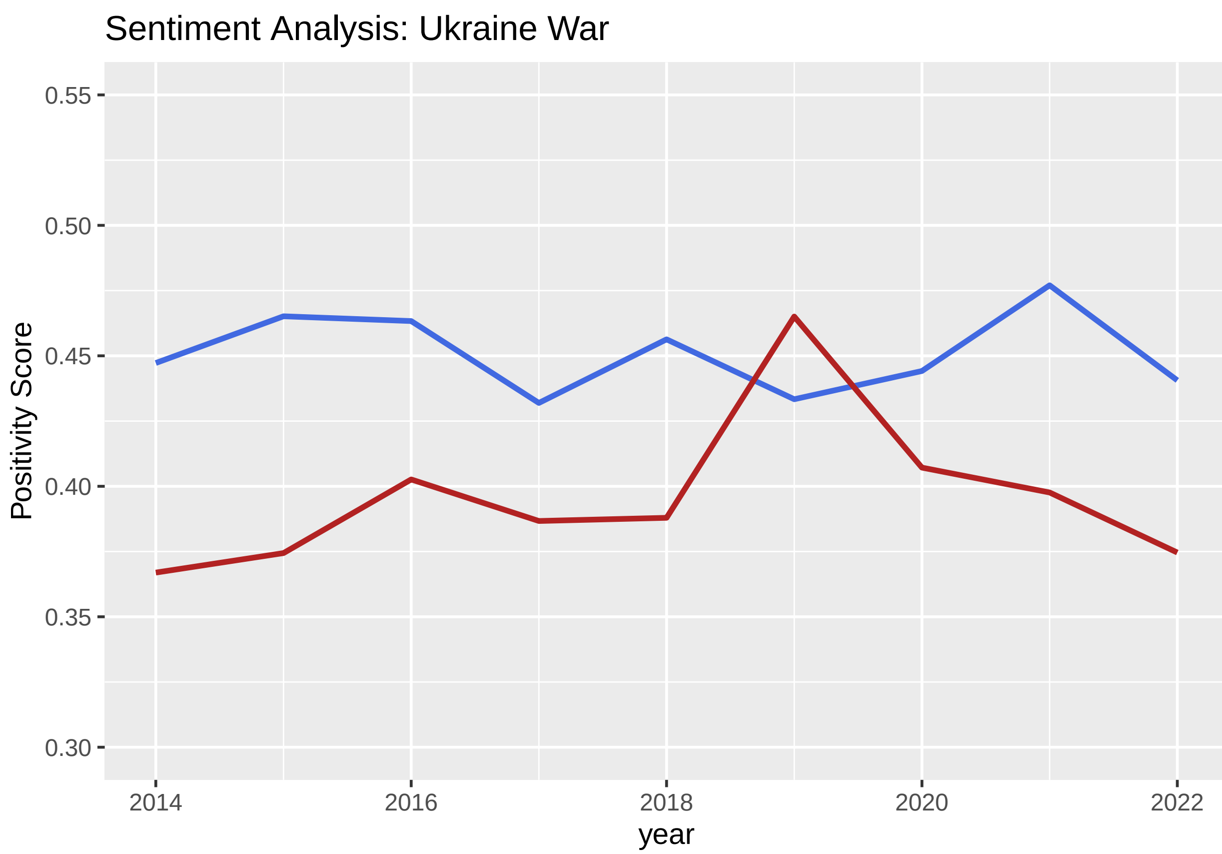

```
afgh_sent_plot <- ggplot()+
  geom_line(afgh_by_year, mapping = aes(x=year, y=pos_neg), size = 1, color = "royalblue")+
  geom_line(nyt_afgh_by_year, mapping = aes(x=year, y=pos_neg, group=1), size = 1, color = "firebrick")
  ylim(0.3, 0.55)+
  labs(title = "Sentiment Analysis: Afghanistan War", y="Positivity Score")
afgh_sent_plot
```




```
syria_sent_plot <- ggplot()+
  geom_line(syria_by_year, mapping = aes(x=year, y=pos_neg), size = 1, color = "royalblue")+
  geom_line(nyt_syria_by_year, mapping = aes(x=as.numeric(year), y=pos_neg, group=1), size = 1, color =
  ylim(0.3, 0.55)+
  labs(title = "Sentiment Analysis: Syria War", y="Positivity Score")
syria_sent_plot
```




```r
ukraine_sent_plot <- ggplot()+
  geom_line(ukraine_by_year, mapping = aes(x=year, y=pos_neg), size = 1, color = "royalblue")+
  geom_line(nyt_ukraine_by_year, mapping = aes(x=year, y=pos_neg, group=1), size = 1, color = "firebric
  ylim(0.3, 0.55)+
  labs(title = "Sentiment Analysis: Ukraine War", y="Positivity Score")
ukraine_sent_plot
```



In general one can say that the Guardian has a more positive broadcasting style then the New York Times. This is consistent over all the wars and only in certain years has the New York Times a higher average Sentiment score than the Guardian. But it is noticeable that the difference is larger and even more consistent in the war in Syria. In comparison, in the Afghanistan war (in which the USA was the leading party) the New York Times has in three years a higher positivity score than the Guardian and also the fluctuation is very high. The difference between the newspapers could be a result of a broadcasting style or also of political agendas. The results from the New york Times suggests that there is something to further look into. But the differences could also be the result of language norms in the two different countries.

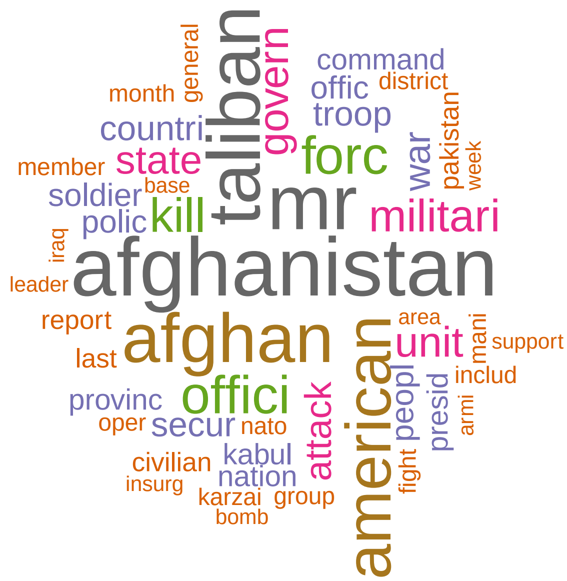
Word Frequency

```
load("Data/afgh_dfm.RData")
load("Data/syria_dfm.RData")
load("Data/ukraine_dfm.RData")
load("Data/nyt_afgh_dfm.RData")
load("Data/nyt_syria_dfm.RData")
load("Data/nyt_ukraine_dfm.RData")
```

```
afgh_textplot <- textplot_wordcloud(afgh_dfm,
                   max_words = 50,
                   random_order = FALSE,
                   rotation = .3,
                   color = RColorBrewer::brewer.pal(8, "Dark2"))
```

Afghanistan

```
nyt_afgh_textplot <- textplot_wordcloud(nyt_afgh_dfm,
                     max_words = 50,
                     random_order = FALSE,
                     rotation = .3,
                     color = RColorBrewer::brewer.pal(8, "Dark2"))
```

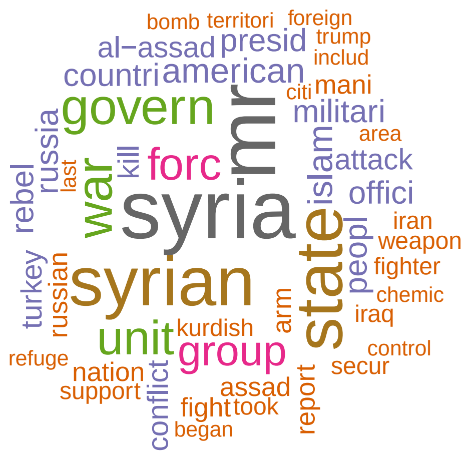


Here is a clear difference noticeable between the two newspapers. The New York Times uses the words "Afghanistan" and especially "Taliban" much more often then the Guardian. This is to be expected since the USA (the origin country of the New York Times) began the war in Afghanistan after the 9/11 terrorist attack executed by the Taliban. This led to the Taliban being the concept of the enemy.

```
syria_textplot <- textplot_wordcloud(syria_dfm,
                    max_words = 50,
                    random_order = FALSE,
                    rotation = .3,
                    color = RColorBrewer::brewer.pal(8, "Dark2"))
```

Syria

```
nyt_syria_textplot <- textplot_wordcloud(nyt_syria_dfm,
                   max_words = 50,
                   random_order = FALSE,
                   rotation = .3,
                   color = RColorBrewer::brewer.pal(8, "Dark2"))
```



There are no real notable differences between the newspapers visible.

```
ukraine_textplot <- textplot_wordcloud(ukraine_dfm,
                    max_words = 50,
                    random_order = FALSE,
                    rotation = .3,
                    color = RColorBrewer::brewer.pal(8, "Dark2"))
```

Ukraine

```r
nyt_ukraine_textplot <- textplot_wordcloud(nyt_ukraine_dfm,
                   max_words = 50,
                   random_order = FALSE,
                   rotation = .3,
                   color = RColorBrewer::brewer.pal(8, "Dark2"))
```

There are no real notable differences between the newspapers visible.

Over all Word Frequencies

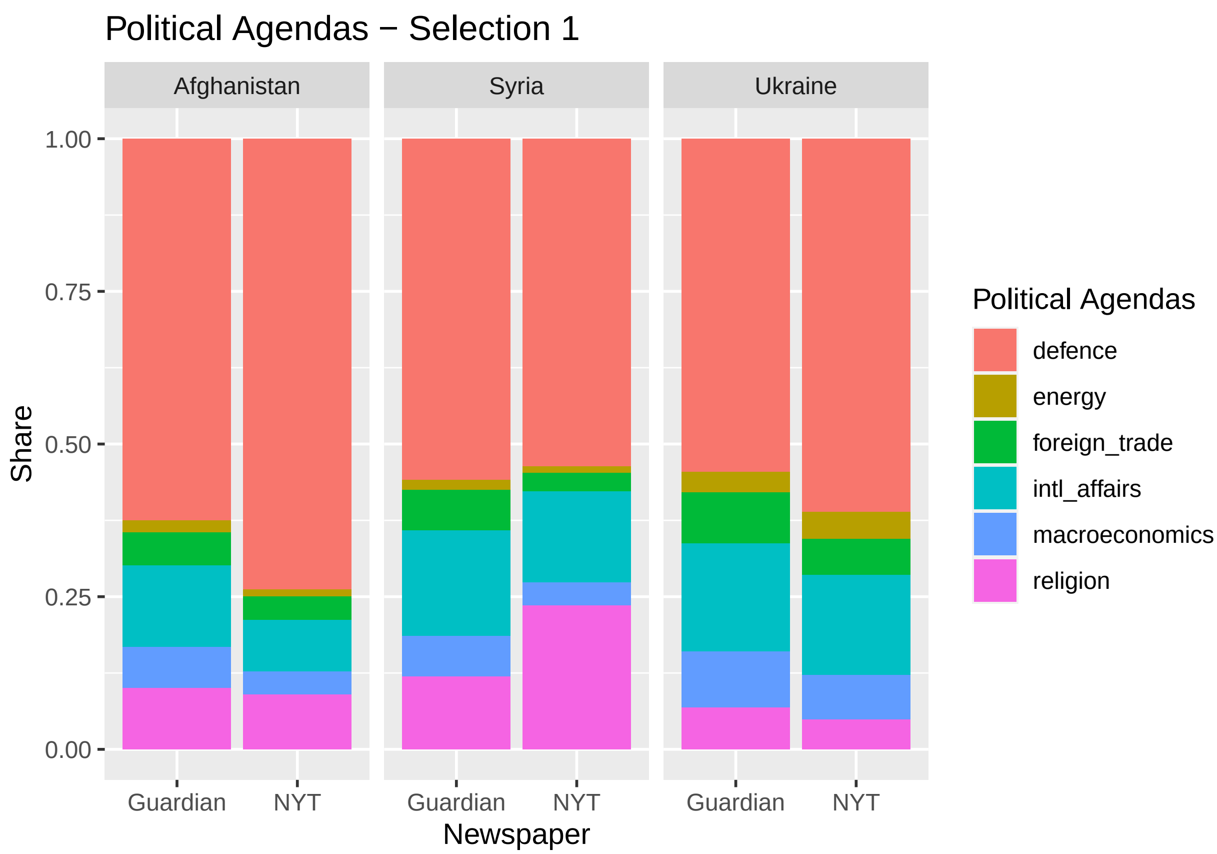
It is difficult here to make a general statement. The noticeable differences between the newspaper could be from the involvement of the country of origin or also from language norms. It is difficult to spot if a difference really comes from a different style of news portrayal. One consistent difference is that the Guardian uses the word "war" much more often then the New York Times. On the other hand it is noticeable that the New York Times very often uses the word "mr" which would suggest a more person-related coverage. But I would argue that that's a result of different language norms in the UK and US.

Policy Agendas

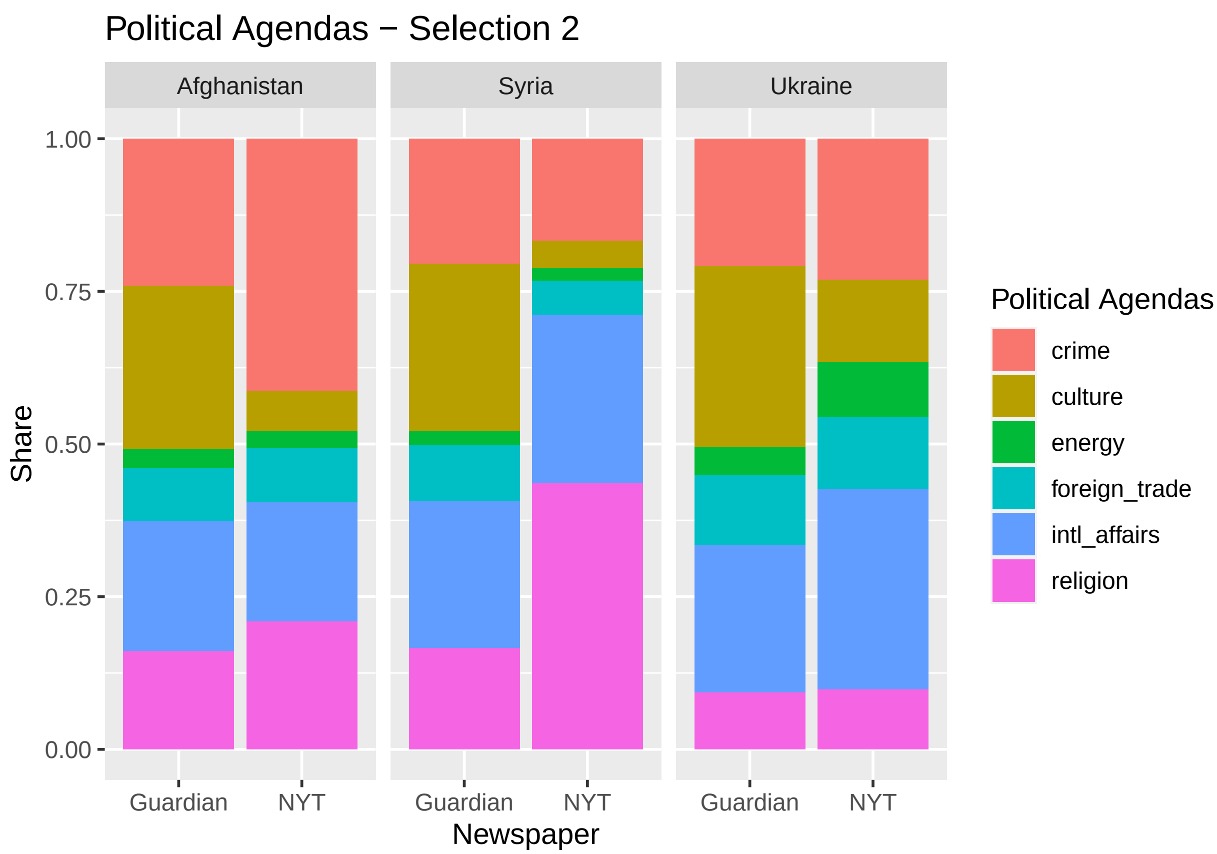
```
load("Data/afgh_pol.RData")
load("Data/syria_pol.RData")
load("Data/ukraine_pol.RData")
load("Data/nyt_afgh_pol.RData")
load("Data/nyt_syria_pol.RData")
load("Data/nyt_ukraine_pol.RData")
```

```
afgh_pol_total <- afgh_pol %>% select(-year) %>%
  colSums(.) %>%
  enframe()
afgh_pol_total$war <- "Afghanistan"
afgh_pol_total$newspaper <- "Guardian"
```

```
syria_pol_total <- syria_pol %>% select(-year) %>%
  colSums(.) %>%
  enframe()
syria_pol_total$war <- "Syria"
syria_pol_total$newspaper <- "Guardian"

ukraine_pol_total <- ukraine_pol %>% select(-year) %>%
  colSums(.) %>%
  enframe()
ukraine_pol_total$war <- "Ukraine"
ukraine_pol_total$newspaper <- "Guardian"

nyt_afgh_pol_total <- nyt_afgh_pol %>% select(-year) %>%
  colSums(.) %>%
  enframe()
nyt_afgh_pol_total$war <- "Afghanistan"
nyt_afgh_pol_total$newspaper <- "NYT"

nyt_syria_pol_total <- nyt_syria_pol %>% select(-year) %>%
  colSums(.) %>%
  enframe()
nyt_syria_pol_total$war <- "Syria"
nyt_syria_pol_total$newspaper <- "NYT"

nyt_ukraine_pol_total <- nyt_ukraine_pol %>% select(-year) %>%
  colSums(.) %>%
  enframe()
nyt_ukraine_pol_total$war <- "Ukraine"
nyt_ukraine_pol_total$newspaper <- "NYT"

pol_total <- rbind(afgh_pol_total, syria_pol_total, ukraine_pol_total,
                   nyt_afgh_pol_total, nyt_syria_pol_total, nyt_ukraine_pol_total)


pol_total_plot <- pol_total %>%
  filter(name %in% c("defence", "energy", "foreign_trade", "intl_affairs", "macroeconomics", "religion"
  ggplot(aes(x=newspaper, y=value, fill = name))+
  geom_bar(stat = "identity",
           position = "fill")+
  facet_grid(~ war)+
  labs(title = "Political Agendas - Selection 1", y="Share", x="Newspaper")+
  scale_fill_discrete(name = "Political Agendas")
pol_total_plot
```




```r
pol_total_plot_2 <- pol_total %>%
  filter(name %in% c("foreign_trade", "intl_affairs", "crime", "religion", "culture", "energy")) %>%
  ggplot(aes(x=newspaper, y=value, fill = name))+
  geom_bar(stat = "identity",
           position = "fill")+
  facet_grid(~ war)+
  labs(title = "Political Agendas - Selection 2", y="Share", x="Newspaper")+
  scale_fill_discrete(name = "Political Agendas")
pol_total_plot_2
```



The comparison shows that there are some differences between the two newspapers. For the Afghanistan war the New York Times seems to concentrate much more on the "defence" part than the Guardian, the same holds for the war in Ukraine. The second selection shows probably the most consistent difference between the two: culture. Over all three wars the Guardian reports much more about the culture aspect. Other major differences are that the New York Times covers more crime in the Afghanistan war and a lot more religion in the Syrian war.

When comparing the wars there are not many clear differences. The differences get mainly concealed by the differences between the newspapers. Nevertheless there are some notable results. Defence is more covered in the Afghanistan war then in the other two, Energy seems to be more important for the war in Ukraine and religion is a larger part in the Syrian war.

Conclusion

Over all there were some noticeable differences especially between the two newspapers. But one problem is consistent over all the analysis methods. It is very difficult to say if the differences in the result come from the differences between the two publishers or from the differences between the two origin countries. To control for that part I would need to look at other newspapers from the two countries to check how newspaper within a country differ from each other. In further research it would also be interesting to dive deeper into each one of the analysis methods. For example would it be interesting to see how the sentiment changes when filtering out articles which contain specific words like "America" or "Russia".

Another important aspect is the data gathering. I chose to use the simple search words "Afghanistan war", "Syria war" and "Ukraine war". A completely unrelated difference between the two newspapers could be in the search algorithm of their API. I don't know how broadly they decide if an article matches a search term or not. So one newspaper could include a much broader variety of articles. Such technical differences could change the outcome of this research.

In conclusion, no real interpretive statement can be made. But this work has shown that there is something to look for and it can inspire further research to get to the bottom of the original question of this paper.