

Geopolitical Changes in the UNSC – Is China’s growing global ambitions reflected in its communication in the UNSC?

Erik Schulte

July 7th 2022

Introduction

References are to be made as follows: Fama and French (1997, 33) and Grinold and Kahn (2000) Such authors could also be referenced in brackets (Grinold and Kahn 2000) and together Grinold and Kahn (2000). Source Theory Part...

Based on the theory of China’s potential shift in policy objectives in the UNSC, I now conduct the empirical evaluation. For this purpose, I first describe the data preparation process. Secondly, I conduct the actual computer assisted text analysis which presents several measures and examines the hypotheses.

Data

First of all, I present all relevant data sources and explain essential features to get a proper overview. Afterwards, I explain the process of loading the data into R and how we process the raw text corpus, to obtain meaningful results.

The dataset comprises UN Security Council debates between January 1995 and December 2020 and was downloaded from Schoenfeld et al. (2019). The official meeting protocols are split into distinct speeches. For every speech, metadata regarding the speaker, the speaker’s nation or affiliation, and the speaker’s role in the meeting is given. The topic of the meeting is also given. In total, the corpus contains 82,165 speeches extracted from 5,748 meeting protocols. Schoenfeld et al. (2019) also provide a codebook with the exact description of their workflow how to set-up the database.

The data are split in to a file containing all information about the UNSC meetings, one file about the metadata of the speeches and one file containing the whole text of the speeches.

Data Sources

For our empirical analysis we use three main data sources. The first two data sources are used for the text analysis and the building of the subgroups. Data on the speeches of the UNSC were obtained from Schönfeld et al. (2019). It contains 82165 speeches over the years 1995 until 2017.

The other two data sources are word dictionaries for the sentiment analysis. It is crucial for the sentiment analysis, that the dictionaries are suitable for the specific context of analysis. For example, a word list developed for sociology may not be a good fit to analyze text data from for finance, and vice versa. The selection of the right word lists is therefore of uttermost importance to get meaningful and robust results Grimmer and Stewart (2013, 274–75). Hence, I carefully choose only certain categories from the dictionaries which are less prone to missclassification. Two wordlists namely Military and Cooperation are from the most

widely used standard dictionary, the Harvard General Inquirer (see Hall (2019)). The other four wordlists: Positive, Uncertainty, Strong Modal and Weak Modal are from “Documentation for the LoughranMcDonald_MasterDictionary”. Word categories that were specially adapted to the financial context by Loughran and McDonald (2011), such as the negative wordlist, are neglected in my analysis.

Data preparation

In the next step, I import the speech data into R from the data folder of my project file. To be able to process larger amounts of text, I need special packages in R. These include the `quanteda`, `quanteda.textplot` and `quanteda.textstats`. Other libraries like `tidyverse`, `magrittr`, `dyplr`, `ggplot2`, `readtext`, and `kableExtra` are used to manipulate the data and plot it. We also set a seed, so that our results are reproducible.

```
#load required packages
library(tidyverse)
library(readtext)
library(quanteda)
library(quanteda.textplots)
library(quanteda.textstats)
library(ggplot2)
library(magrittr)
library(gdata)
library(kableExtra)
library(tidytext)
library(dplyr)

#load the UN Security Council Debates dataset
load("data/dataset.RData")
load("data/docs.RData")

#set a seed for reproducibility
set.seed(2333)

# View data
#head(meta_meetings)
#head(meta_speeches)
```

I then rename the filename column of the data frame containing the metadata of the speeches into `doc_id` in order to have a common identifier for merging with the actual text data from the speeches (`raw_docs`).

```
##rename common column to merge the content of the speeches
meta_speeches <- meta_speeches %>%
  rename(doc_id = filename) |>
  filter(participanttype != "The President")

## merge dataset raw with speeches
meta_speeches <- merge(meta_speeches, raw_docs, by = 'doc_id')
##
##how to do it after a specific column???
#add_column(meta_speeches, raw_docs[c("text")], .after = "filename")
# meta_speeches |>
# summarize(length_speeches = mean(sentences))
# summarize(mean_tokens = mean(tokens))
# summarize(mean_unique_tokens = mean(types))
# meta_speeches |>
```

```
# group_by(country)
# count(as.numeric(meta_speeches$sentences))
# arrange(desc(n))
#
# str(meta_speeches)
```

In the next step I want to make a corpus from my data. The command `corpus` comes from the `quanteda` package. Hence, the text column of the `meta_speeches` dataframe are now a corpus. I filter out speeches from the president of the UNSC, as he or she mostly speaks on behalf of the organization and not for their respective country. This reduces the number of speeches by about 30,000 to 50,933.

```
#as_corpus_frame(meta_speeches$text, filter = NULL, row.names = NULL)
corp_meta_speeches <- corpus(meta_speeches, text_field = "text")
summary(corp_meta_speeches, 1)
```

```
## Corpus consisting of 50933 documents, showing 1 document:
```

```
##
##              Text Types Tokens Sentences speech
## UNSC_1995_SPV.3487_spch002.txt    575   1746      45      2
##              country speaker participanttype role_in_un spv
## Bosnia And Herzegovina Mr. Sacirbey          Guest      3487
##              basename
## UNSC_1995_SPV.3487
##
##              topic
## Federal Republic of Yugoslavia (Serbia and Montenegro) - Sanctions
##              date year month day types tokens sentences
## 12 January 1995 1995      1 12    575   1746      45
##
##              topic2
## Items relating to the situation in the former Yugoslavia
##              subtopic agenda_item1 agenda_item2 agenda_item3
## Bosnia and Herzegovina Europe Former Yugoslavia Bosnia and Herzegovina
## decision
##      <NA>
```

```
corp_meta_speeches <- corpus_subset(corp_meta_speeches, participanttype != "The President")
##
#is_corpus_frame(corp_meta_speeches$text)
```

I also create a unique corpus just for the speeches held by China. I do this using the `corpus_subset` command from my previously defined corpus. I also take a subset of the `meta_speeches` dataframe with only the speeches from China. In total there are 3564 speeches by China in the dataframe, a total share of 3.82 % of all speeches. So there appear to be 1619 speeches from China as participant type president. So, the speeches where china speaks on behalf of the president are excluded.

```
#How do I assign the speeches to the columns (now the speeches are in the same order as in the document

corp_China <- corpus_subset(corp_meta_speeches, country == "China")
Chinaspeeches <- subset.data.frame(meta_speeches, country == "China")
share_CHN_speeches <- 1945/50933
##
#only creates value
#Germany <- meta_speeches$country == "Germany"
```

```
##make country names small (something wrong)
#chinaspeeches_lower <- corpus_subset(corp_meta_speeches, tolower(country) %in% tolower(corp_China))

## access to document level variables of the China corpus
head(docvars(corp_China))
```

```
##      speech country      speaker participanttype role_in_un      spv
## 1      30   China Mr. Wang Xuexian      Mentioned      3487
## 2       4   China Mr. Wang Xuexian      Mentioned      3489
## 3       7   China Mr. Wang Xuexian      Mentioned      3492
## 4      11   China Mr. Li Zhaoxing      Mentioned      3494
## 5      13   China Mr. Li Zhaoxing      Mentioned      3496
## 6      16   China Mr. Wang Xuexian      Mentioned      3499Resumption1
##
##      basename
## 1      UNSC_1995_SPV.3487
## 2      UNSC_1995_SPV.3489
## 3      UNSC_1995_SPV.3492
## 4      UNSC_1995_SPV.3494
## 5      UNSC_1995_SPV.3496
## 6 UNSC_1995_SPV.3499Resumption1
##
##      topic
## 1 Federal Republic of Yugoslavia (Serbia and Montenegro) - Sanctions
## 2      Liberia
## 3      Agenda for Peace
## 4      Mozambique
## 5      Haiti
## 6      Angola
##
##      date year month day types tokens sentences
## 1 12 January 1995 1995      1 12      287      707      22
## 2 13 January 1995 1995      1 13      202      432      16
## 3 18 January 1995 1995      1 18      670     1948      69
## 4 27 January 1995 1995      1 27      261      553      18
## 5 30 January 1995 1995      1 30      197      373      14
## 6  8 February 1995 1995      2  8      358      902      28
##
##      topic2
## 1 Items relating to the situation in the former Yugoslavia
## 2      The situation in Liberia
## 3      Items relating to an Agenda for Peace
## 4      The situation in Mozambique
## 5      The question concerning Haiti
## 6      The situation in Angola
##
##      subtopic agenda_item1      agenda_item2
## 1      Bosnia and Herzegovina      Europe Former Yugoslavia
## 2      The situation in Liberia      Africa      Liberia
## 3 Items relating to an Agenda for Peace      Thematic      Agenda for peace
## 4      The situation in Mozambique      Africa      Mozambique
## 5      The question concerning Haiti      Americas      Haiti
## 6      The situation in Angola      Africa      Angola
##
##      agenda_item3 decision
## 1 Bosnia and Herzegovina      <NA>
## 2      Liberia      <NA>
## 3      Thematic      <NA>
## 4      Mozambique      <NA>
```

```
## 5          Haiti    <NA>
## 6          Angola    <NA>
```

```
##extract them the document level variables
#docvars(corp_China, field = "speaker")

#corp_China$year

##create new variables (does not work yet)
#corp_China$avg_speech_length <- (mean(corp_China$sentences))
#avg_speech_length
```

As a next step, I separate the whole speeches into sentences in tidy format. One column for every sentence of a speech. For that I use the `unnest_tokens` function from the `tidytext` package. This lets the the number of observations grow to 1,767,696 million. The problem here is that now after every salutation like “Mr.” so after every point the function creates a new sentence. When we create a dataframe for every word as an observation, the number of observation grows to 43,534,652 million. Therefore, we must clean the data further before using them for analysis.

```
##separates by default into words all speeches from raw docs (here by sentences) variable after tokens
tidy_raw_sentences <- meta_speeches %>%
  unnest_tokens(sentences_content, text, token = "sentences" )

##separate into words a new variable each word per speech (gives us 47,4 Mio obs.)
tidy_raw_words <- meta_speeches |>
  unnest_tokens(word, text)
```

The `tidytext` package also come with a list of stopwords – words that are not meaningful and that we want to exclude from the analysis. The stopwords list contains 1149 words. Our dataframe reduces to 19,623,316 million observations – by more than a half. This allows me to start with a first analysis. I can plot the most frequent words used in the speeches of the UNSC. For that, I plot the most common words that were used more than 80,000 times in total. The threshold of 65,000 is subjectively chosen and is based on the graphical aesthetics.

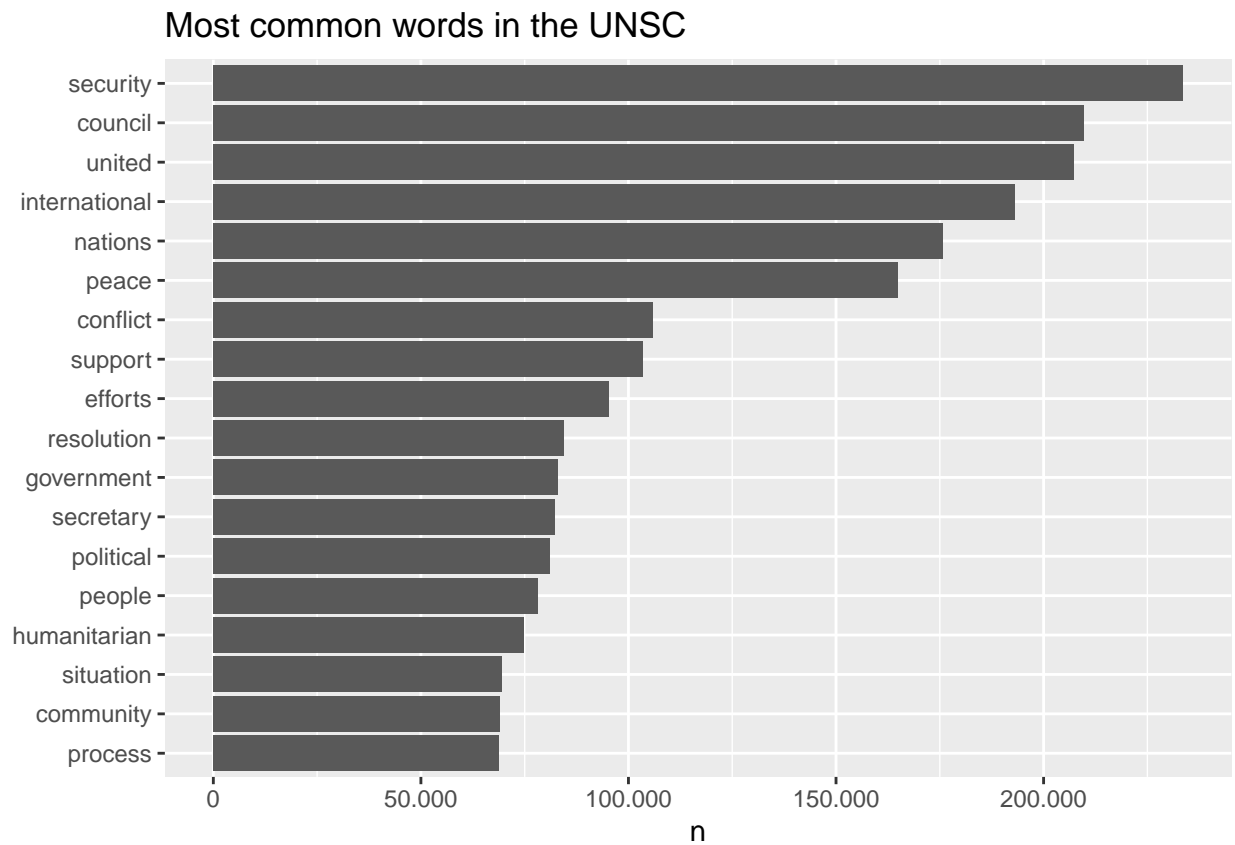
```
##remove redundant words (reduces from 47,6 Mio words to 21,4 Mio words)
data(stop_words) ##this comes with a package (a list of unuseful words)

tidy_raw_words <- tidy_raw_words %>%
  anti_join(stop_words)

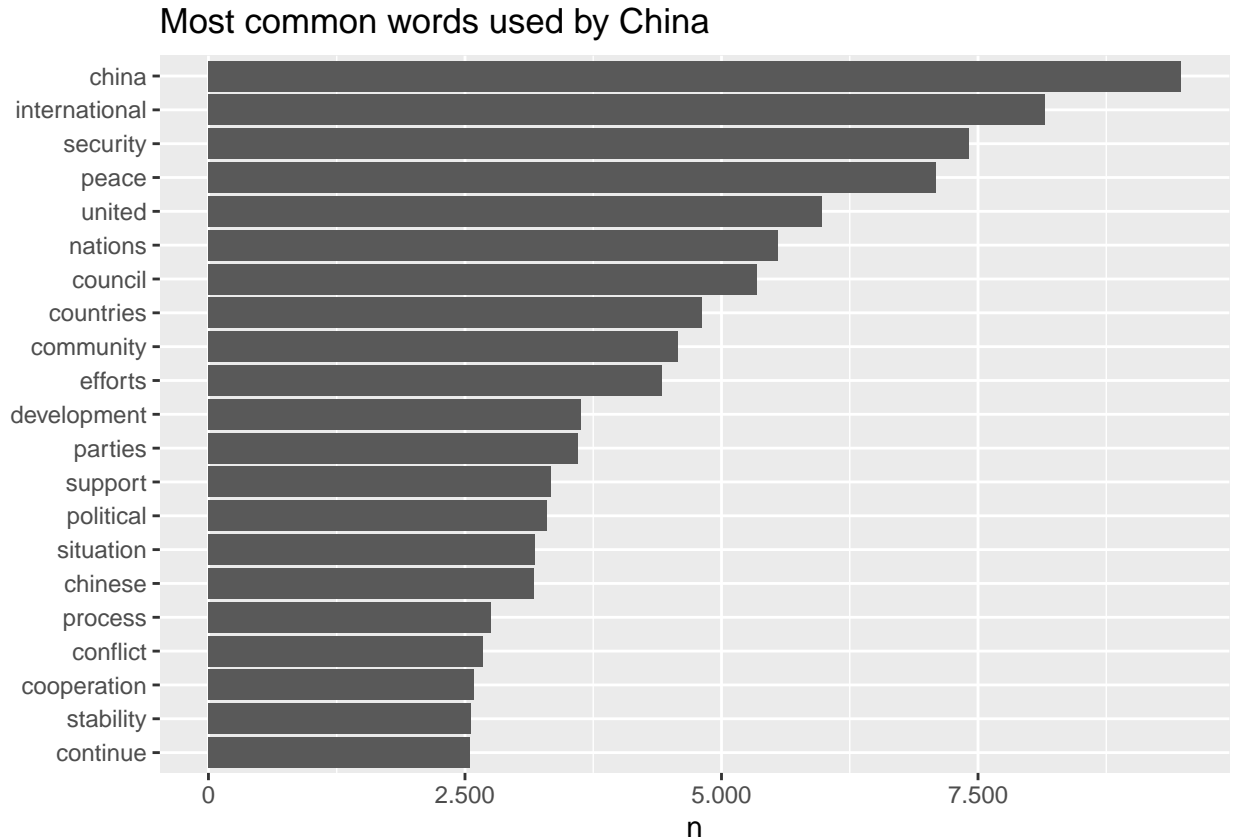
##count words, after redundant words were removed (only meaningful words)
# tidy_raw_words %>%
#   count(word, sort = TRUE)
```

A first visual inspection of the most commonly used words

We can see from the graph that council, security and united are the most frequently used words. This gives are first overview of the most used words.



I then apply the same analysis for the dataframe from the Chinese speeches.



Frequency terms

Next, I calculate frequency terms. The function `get_freqs` creates a list of the frequency of terms that appear in each speech. The function takes the respective corpus as an input and returns the frequencies in the form of a data frame. In the first step, it creates tokens from the text documents. Tokens are a sequence of elementary lexical components, in our case words. Punctuation, separators and stop words like “he,” “do,” or “if” are excluded as they have no meaning for the content and therefore represent no real added value for the analysis. The `dfm()` function is then applied to the tokens to create the document feature matrix (dfm). As the dfm is a very sparse matrix containing the documents as rows and the terms as columns, I aggregate the data over all documents and sort the frequency of the terms in a descending order using the `textstat_frequency()` command. In addition, a special feature of the `get_freqs` function is the weighting parameter, which is set to `FALSE` per default. If it is being activated the frequencies are weighted according to the term-frequency inverse document frequency (tf.idf) weighting scheme. The theory behind this term weighting scheme is further explained in section B.3. In the end we apply the `get_dfm` function to the two corpora and save the result in the variable `freqs_all` and `freqs_CHN` respectively.

i have to remove (“interpretation from Chinese first actually)

```
#calculate the frequency of terms that appear in each speech. It also creates a document feature matrix
#frequency of each word in the corpus
# function takes the respective corpus as an input and returns the frequencies in the form of a data fr
```

```

get_freqs <- function(corp_meta_speeches, weighting = TRUE){
tokens <- tokens(corp_meta_speeches, remove_punct = T, remove_separators = T, include_docvars = T)
tokens <- tokens_remove(tokens, stopwords("en"))
dfm <- tokens %>% dfm()
freqs <- textstat_frequency(dfm)

if(weighting == TRUE){
freqs$term_frequency <- freqs$frequency / sum(freqs$frequency)
freqs$inverse_doc_freq <- log10( length(corp_meta_speeches) / freqs$docfreq )
freqs$frequency <- freqs$term_frequency*freqs$inverse_doc_freq
}
return(freqs)
}

freqs_CHN <- get_freqs(corp_China,weighting=TRUE)
freqs_all <- get_freqs(corp_meta_speeches,weighting = TRUE)

##why is the frequency in general a bit lower than in the tasks before with the plot where a plot a gray

#Mr. und Mrs. daran kann man vielleicht etwas zur Frauenquote sagen

```

We then get a new dataframe with each word, the frequency (percentage share of appearance of all words), the rank based on the frequency, the total document frequency of the words, the term_frequency and the inverse_term_frequency.

Sentiment Analysis

For the sentiment analysis, I have to get dictionaries for my purpose into R. I choose to do a dictionary-based approach as I am a beginner in text analysis. In accordance with processing of the speech data, we lower all dictionary terms so that the matching function is case sensitive. Two wordlists namely Military and Cooperation are from the most widely used standard dictionary, the Harvard General Inquirer Hall (2019). I manually collected this word list. It was difficult to obtain due to a restricted access. I create the new dataframe dict with these two categories after making all words to lower case letters.

```

library(readxl)
GI_dic <- read_excel("~/ownCloud/Uni Göttingen/Stellenbosch University/Data Science Methods/Github Proj

GI_dic$military <- tolower(GI_dic$military) #make words to lower case in military columns
GI_dic$powercoop <- tolower(GI_dic$powercoop) #mark words to lower case in powercoop column
colnames(GI_dic) <- tolower(colnames(GI_dic))

dict <- GI_dic |>
  select(military, powercoop)

```

The second dictionary I am using is the one by “Documentation for the LoughranMcDonald_MasterDictionary”. From this I append the wordlists “Positive,”Uncertainty”, “Strongmodal” and “Weakmodal” to my dict.


```

loughran <- read_excel("~/ownCloud/Uni Göttingen/Stellenbosch University/Data Science Methods/Github Pro

colnames(loughran) <- tolower(colnames(loughran))
# Select the following 4 categories for my analysis, exclude negative inter alia
loughran <- loughran[c("positive","uncertainty","strongmodal","weakmodal")]
# make all words to lower case
loughran$positive <- tolower(loughran$positive)
loughran$uncertainty <- tolower(loughran$uncertainty)
loughran$strongmodal <- tolower(loughran$strongmodal)
loughran$weakmodal <- tolower(loughran$weakmodal)

#cbindX = column-binds objects with different number of rows.
dict <- cbindX(dict, loughran)
colnames(dict) <- c("Military","Cooperation","Positive","Uncertainty","Strongmodal",
"Weakmodal")

```

The second important function called `sentiment_analysis` uses this `dict` data frame together with the previously created term frequency tables as inputs. It then calculates a proportional count of each category to measure the tone of the language and outputs the result as a data frame. To do so, the function initializes a storage table for our results named `data`. The column names of the result table are set to the categories contained in the dictionary table. We then iterate through the column names (i.e. the categories) to match the words of each category with the term frequency tables. The resulting data frame `join_1` contains only the words, that are in both data frames (this is called an inner join). We then calculate the share of words for each category from the overall frequency. The same is applied to the second term frequency table and then saved to the result table.

```

sentiment_analysis <- function (freqs_all, freqs_CHN, dict){
  data <- data.frame(matrix(NA,2,dim(dict)[2])) #data is the storage table
  colnames(data) <- colnames(dict)

  for (i in colnames(data)){
    join_1 <- freqs_all %>% inner_join(dict, by= c("feature" = i))
    a <- sum(join_1$frequency)/sum(freqs_all$frequency)*100

    join_2 <- freqs_CHN %>% inner_join(dict, by= c("feature" = i))
    b <- sum(join_1$frequency)/sum(freqs_CHN$frequency)*100

    data[i] <- rbind(a,b)
  }
  return(data)
}

```

```

##also create a tidy dictionary, we I have 863 words with 5 sentiment categories in total
tidy_dict <- dict %>% tidyr::gather("Sentiment", value, 1:6) |>
  rename(word = value) |>
  drop_na()

```

References

- “Documentation for the LoughranMcDonald_MasterDictionary.” https://www3.nd.edu/~mcdonald/Word_Lists_files/Documentation/Documentation_LoughranMcDonald_MasterDictionary.pdf.
- Fama, Eugene F, and Kenneth R French. 1997. “Industry Costs of Equity.” *Journal of Financial Economics* 43 (2): 153–93.
- Grimmer, Justin, and Brandon M Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21 (3): 267–97.
- Grinold, Richard C, and Ronald N Kahn. 2000. “Active Portfolio Management.”
- Hall, William James. 2019. “Harvard Iv-4 General Inquire Dictionary.” 2019. <http://www.wjh.harvard.edu/~inquirer/homecat.htm>.
- Schoenfeld, Mirco, Steffen Eckhard, Ronny Patz, Hilde van Meegdenburg, and Antonio Pires. 2019. “The UN Security Council Debates.” Harvard Dataverse. <https://doi.org/10.7910/DVN/KGVSYH>.
- Schönfeld, Mirco, Steffen Eckhard, Ronny Patz, and Hilde van Meegdenburg. 2019. “The UN Security Council Debates 1995-2017.” *arXiv Preprint arXiv:1906.10969*.