Essay\_markdown\_final

River Kim

2024-04-07

# Set up

Data importing (Following Course Book “Assessment Data”)

library(ggplot2)  
library(plyr)  
library(gdata)  
library(stringr)  
library(data.table)

## Prep Osnabrugge et al.  
data = fread("/Users/garamkim/Downloads/dataverse\_files/uk\_data.csv", encoding="UTF-8")  
data$date = as.Date(data$date)  
  
#Create time variable  
data$time= NA  
data$time[data$date>=as.Date("2001-01-01") & data$date<=as.Date("2001-06-30")] = "01/1"  
data$time[data$date>=as.Date("2001-07-01") & data$date<=as.Date("2001-12-31")] = "01/2"  
data$time[data$date>=as.Date("2002-01-01") & data$date<=as.Date("2002-06-30")] = "02/1"  
data$time[data$date>=as.Date("2002-07-01") & data$date<=as.Date("2002-12-31")] = "02/2"  
data$time[data$date>=as.Date("2003-01-01") & data$date<=as.Date("2003-06-30")] = "03/1"  
data$time[data$date>=as.Date("2003-07-01") & data$date<=as.Date("2003-12-31")] = "03/2"  
data$time[data$date>=as.Date("2004-01-01") & data$date<=as.Date("2004-06-30")] = "04/1"  
data$time[data$date>=as.Date("2004-07-01") & data$date<=as.Date("2004-12-31")] = "04/2"  
data$time[data$date>=as.Date("2005-01-01") & data$date<=as.Date("2005-06-30")] = "05/1"  
data$time[data$date>=as.Date("2005-07-01") & data$date<=as.Date("2005-12-31")] = "05/2"  
data$time[data$date>=as.Date("2006-01-01") & data$date<=as.Date("2006-06-30")] = "06/1"  
data$time[data$date>=as.Date("2006-07-01") & data$date<=as.Date("2006-12-31")] = "06/2"  
data$time[data$date>=as.Date("2007-01-01") & data$date<=as.Date("2007-06-30")] = "07/1"  
data$time[data$date>=as.Date("2007-07-01") & data$date<=as.Date("2007-12-31")] = "07/2"  
data$time[data$date>=as.Date("2008-01-01") & data$date<=as.Date("2008-06-30")] = "08/1"  
data$time[data$date>=as.Date("2008-07-01") & data$date<=as.Date("2008-12-31")] = "08/2"  
data$time[data$date>=as.Date("2009-01-01") & data$date<=as.Date("2009-06-30")] = "09/1"  
data$time[data$date>=as.Date("2009-07-01") & data$date<=as.Date("2009-12-31")] = "09/2"  
data$time[data$date>=as.Date("2010-01-01") & data$date<=as.Date("2010-06-30")] = "10/1"  
data$time[data$date>=as.Date("2010-07-01") & data$date<=as.Date("2010-12-31")] = "10/2"  
data$time[data$date>=as.Date("2011-01-01") & data$date<=as.Date("2011-06-30")] = "11/1"  
data$time[data$date>=as.Date("2011-07-01") & data$date<=as.Date("2011-12-31")] = "11/2"  
data$time[data$date>=as.Date("2012-01-01") & data$date<=as.Date("2012-06-30")] = "12/1"  
data$time[data$date>=as.Date("2012-07-01") & data$date<=as.Date("2012-12-31")] = "12/2"  
data$time[data$date>=as.Date("2013-01-01") & data$date<=as.Date("2013-06-30")] = "13/1"  
data$time[data$date>=as.Date("2013-07-01") & data$date<=as.Date("2013-12-31")] = "13/2"  
data$time[data$date>=as.Date("2014-01-01") & data$date<=as.Date("2014-06-30")] = "14/1"  
data$time[data$date>=as.Date("2014-07-01") & data$date<=as.Date("2014-12-31")] = "14/2"  
data$time[data$date>=as.Date("2015-01-01") & data$date<=as.Date("2015-06-30")] = "15/1"  
data$time[data$date>=as.Date("2015-07-01") & data$date<=as.Date("2015-12-31")] = "15/2"  
data$time[data$date>=as.Date("2016-01-01") & data$date<=as.Date("2016-06-30")] = "16/1"  
data$time[data$date>=as.Date("2016-07-01") & data$date<=as.Date("2016-12-31")] = "16/2"  
data$time[data$date>=as.Date("2017-01-01") & data$date<=as.Date("2017-06-30")] = "17/1"  
data$time[data$date>=as.Date("2017-07-01") & data$date<=as.Date("2017-12-31")] = "17/2"  
data$time[data$date>=as.Date("2018-01-01") & data$date<=as.Date("2018-06-30")] = "18/1"  
data$time[data$date>=as.Date("2018-07-01") & data$date<=as.Date("2018-12-31")] = "18/2"  
data$time[data$date>=as.Date("2019-01-01") & data$date<=as.Date("2019-06-30")] = "19/1"  
data$time[data$date>=as.Date("2019-07-01") & data$date<=as.Date("2019-12-31")] = "19/2"  
  
data$time2 = data$time  
data$time2 = str\_replace(data$time2, "/", "\_")  
  
data$stage = 0  
data$stage[data$m\_questions==1]= 1  
data$stage[data$u\_questions==1]= 2  
data$stage[data$queen\_debate\_others==1]= 3  
data$stage[data$queen\_debate\_day1==1]= 4  
data$stage[data$pm\_questions==1]= 5

Inspecting data and selecting some parts of data for the research

# Packages  
library(tidyverse)  
library(readr)  
library(tidytext)  
library(quanteda)  
library(textdata)

#Filtering data columns  
colnames(data)

## [1] "id\_speech" "id\_mp"   
## [3] "period" "last\_name"   
## [5] "first\_name" "date"   
## [7] "pm\_questions" "queen\_debate\_day1"   
## [9] "queen\_debate\_others" "m\_questions"   
## [11] "u\_questions" "other\_debate"   
## [13] "leader" "prime\_minister"   
## [15] "senior\_minister" "shadow"   
## [17] "cabinet" "chair"   
## [19] "government" "female"   
## [21] "age" "electoral\_cycle"   
## [23] "party" "linear\_trend"   
## [25] "words" "text"   
## [27] "emotive\_count" "neutral\_count"   
## [29] "emotive\_rhetoric" "emotive\_rhetoric\_log"   
## [31] "emotive\_words" "top\_topic"   
## [33] "anew\_rescaled" "emotive\_rhetoric\_liwc"   
## [35] "positive\_count" "negative\_count"   
## [37] "emotive\_positive" "emotive\_negative"   
## [39] "emotive\_count\_250\_8" "neutral\_count\_250\_8"   
## [41] "emotive\_rhetoric\_250\_8" "emotive\_count\_300\_10"   
## [43] "neutral\_count\_300\_10" "emotive\_rhetoric\_300\_10"  
## [45] "emotive\_count\_a1" "neutral\_count\_a1"   
## [47] "emotive\_rhetoric\_a1" "emotive\_count\_a2"   
## [49] "neutral\_count\_a2" "emotive\_rhetoric\_a2"   
## [51] "time" "time2"   
## [53] "stage"

data <- data %>%  
 select(last\_name, first\_name, date, female, age, party, text)

Defining research data by filtering research words

# Filtering data containing immigration related words  
# Define the keywords to search for  
immig\_words <- c('immigration', 'immigrant', 'asylum')  
visa\_words <- "\\b(UK)?visas?\\b"  
all\_words <- paste0(c(paste0(immig\_words, collapse = "|"), visa\_words), collapse = "|")  
  
# lower case text with keywords  
tidy\_data\_notoken <- data %>%  
 mutate(desc = tolower(text)) %>%  
 filter(grepl(all\_words, desc))  
  
# arrange the data by party and count the number of speech  
tidy\_data\_notoken %>%  
 group\_by(party) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count))

## # A tibble: 9 × 2  
## party count  
## <chr> <int>  
## 1 Conservative 7230  
## 2 Labour 6251  
## 3 Scottish National Party 1104  
## 4 Liberal Democrats 954  
## 5 Others 105  
## 6 Democratic Unionist Party 67  
## 7 Plaud Cymru 63  
## 8 Green 21  
## 9 Ulster Unionist Party 14

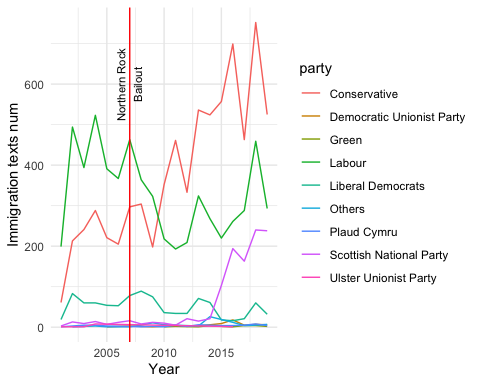
# 1. Word frequency of data

Making ‘Year’ column

# Make 'Year' using 'date'  
tidy\_data\_notoken$Year <- format(as.Date(tidy\_data\_notoken$date), "%Y")  
data$Year <- format(as.Date(data$date), "%Y")  
  
# Count the number of text by year and party  
text\_count <- tidy\_data\_notoken %>%   
 group\_by(Year, party) %>%   
 summarise(n = n(), .groups = 'drop')  
  
text\_count$Year <- as.numeric(as.character(text\_count$Year)) # Make 'Year' Column of numeric character.

A Simple descriptive plot - The number of immigration related texts by party

# Plotting counts of immigration texts by party  
ggplot(text\_count, aes(x = Year, y = n, group = party, color = party)) +  
 geom\_line() +  
 labs(y = "Immigration texts num", x = "Year") +  
 theme\_minimal() +  
 geom\_vline(xintercept = as.numeric(format(as.Date("2007-09-14"), "%Y")), col="red") +  
 annotate("text", x = as.numeric(format(as.Date("2007-09-14"), "%Y")), y = 600, label="Northern Rock\nBailout", angle=90, color = "black", size = 3)

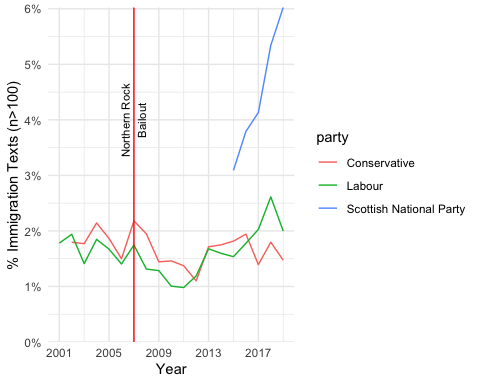


Dividing the number of immigration texts by the total number of texts

# Count the total number of texts in data by year and party  
data\_count <- data %>%  
 group\_by(Year, party) %>%  
 summarise(total\_n = n(), .groups = 'drop')  
  
# Merge the counts and calculate the ratio  
ratio\_count <- merge(text\_count, data\_count, by = c("Year", "party"))  
ratio\_count$ratio <- with(ratio\_count, n / total\_n)  
  
# Only counts more than 100 texts  
filtered\_ratio\_count <- ratio\_count %>%  
 filter(n > 100)  
  
filtered\_ratio\_count$Year <- as.numeric(as.character(filtered\_ratio\_count$Year))

A ratio plot of a number of immigration related texts by total texts

#Plotting the ratios of immigration related texts   
ggplot(filtered\_ratio\_count, aes(x = Year, y = ratio, group = party, color = party)) +  
 geom\_line() +  
 labs(y = "% Immigration Texts (n>100)", x = "Year") +  
 scale\_y\_continuous(labels = scales::percent\_format(), expand = c(0, 0), limits = c(0, NA)) +  
 scale\_x\_continuous(breaks = seq(min(filtered\_ratio\_count$Year), max(filtered\_ratio\_count$Year), by = 4)) +  
 theme\_minimal() +  
 geom\_vline(xintercept = as.numeric(format(as.Date("2007-09-14"), "%Y")), col="red") +  
 annotate("text", x = as.numeric(format(as.Date("2007-09-14"), "%Y")), y = 0.04, label="Northern Rock\nBailout", angle=90, color = "black", size = 3)



Tokenisation and processing stop words

# Tokenisation & removing stop words  
tidy\_data <- tidy\_data\_notoken %>%  
 unnest\_tokens(word, desc) %>%  
 filter(str\_detect(word, "[a-z]")) %>%  
 filter(!word %in% stop\_words$word)  
  
tidy\_data <- tidy\_data %>%  
 arrange(date)  
tidy\_data$order <- 1:nrow(tidy\_data) # Make orders of each word

# Common tokens  
word\_count <- tidy\_data %>%  
 count(word, sort = T)  
  
show(word\_count)

## word n  
## <char> <int>  
## 1: people 27644  
## 2: hon 26615  
## 3: government 26551  
## 4: immigration 19105  
## 5: minister 14229  
## ---   
## 39501: énarques 1  
## 39502: ørsted 1  
## 39503: þæt 1  
## 39504: šefcovic 1  
## 39505: štefan 1

# Common tokens by year  
word\_count\_year <- tidy\_data %>%  
 group\_by(Year) %>%  
 count(word, sort = T)  
  
show(word\_count\_year)

## # A tibble: 226,043 × 3  
## # Groups: Year [19]  
## Year word n  
## <chr> <chr> <int>  
## 1 2018 government 2122  
## 2 2018 people 2060  
## 3 2015 people 2058  
## 4 2018 immigration 2046  
## 5 2004 government 2007  
## 6 2002 people 1883  
## 7 2018 hon 1851  
## 8 2004 hon 1826  
## 9 2002 hon 1792  
## 10 2007 hon 1792  
## # ℹ 226,033 more rows

# Comon tokens by party  
word\_count\_party <- tidy\_data %>%  
 group\_by(party) %>%  
 count(word, sort = T)  
  
show(word\_count\_party)

## # A tibble: 98,986 × 3  
## # Groups: party [9]  
## party word n  
## <chr> <chr> <int>  
## 1 Labour people 12163  
## 2 Conservative government 12117  
## 3 Labour hon 12010  
## 4 Conservative hon 11848  
## 5 Conservative people 11229  
## 6 Labour government 9743  
## 7 Conservative immigration 9001  
## 8 Labour immigration 7336  
## 9 Conservative minister 6290  
## 10 Conservative country 6258  
## # ℹ 98,976 more rows

# 2. Sentiment Analysis Using ‘NRC’ Sentiment Dictionary

## 1)Total sentiment of immigration related texts

Getting NRC sentiment dictionary and tidying data by calculating sentiment ratio

get\_sentiments("nrc")

## # A tibble: 13,872 × 2  
## word sentiment  
## <chr> <chr>   
## 1 abacus trust   
## 2 abandon fear   
## 3 abandon negative   
## 4 abandon sadness   
## 5 abandoned anger   
## 6 abandoned fear   
## 7 abandoned negative   
## 8 abandoned sadness   
## 9 abandonment anger   
## 10 abandonment fear   
## # ℹ 13,862 more rows

# Make nrc sentiment tables of data  
nrc\_data <- tidy\_data %>%  
 inner\_join(get\_sentiments("nrc"), by = "word") %>%  
 count(date, sentiment) %>%  
 spread(key = sentiment, value = n, fill = 0) %>%  
 mutate(ratio = negative / (positive+1)) # Calculating negative/positive ratio. Adding 1 to avoid the 0 denominator

## Warning in inner\_join(., get\_sentiments("nrc"), by = "word"): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 6 of `x` matches multiple rows in `y`.  
## ℹ Row 5657 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

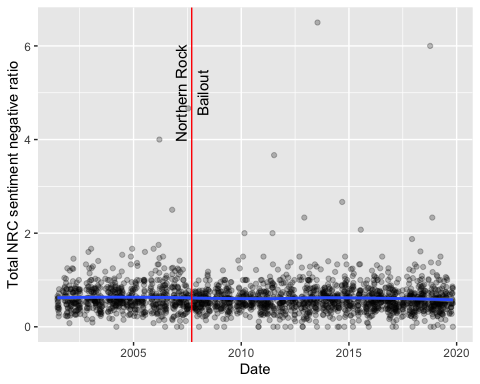
# Check sentiments  
colnames(nrc\_data)

## [1] "date" "anger" "anticipation" "disgust" "fear"   
## [6] "joy" "negative" "positive" "sadness" "surprise"   
## [11] "trust" "ratio"

Plotting the total negative ratio

# Total sentiment of text including immigration related words plot (Using negative ratio)  
nrc\_data %>%  
 ggplot(aes(date, ratio)) +  
 geom\_point(alpha=0.25) +  
 geom\_smooth(method="loess", alpha=0.5) +  
 labs(y = "Total NRC sentiment negative ratio", x = "Date") +  
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col="red") +  
 annotate("text", x = as.Date("2007-09-24"), y = 5, label="Northern Rock\nBailout", angle=90, color = "black", size = 4)

## `geom\_smooth()` using formula = 'y ~ x'

 There is no time when particularly negative emotions are revealed. Rather, negative feelings toward immigrants are less visible after the financial crisis.

## 2) Negative ratio plots by party

Calculating NRC sentiment negative ratios of texts which contain immigration related words *by party*

# Arranging data by party and calculate the ratio of negative sentiment.  
nrc\_data\_party <- tidy\_data %>%  
 inner\_join(get\_sentiments("nrc"), by = "word") %>%  
 group\_by(party, date) %>%  
 count(sentiment) %>%  
 spread(key = sentiment, value = n, fill = 0) %>%  
 mutate(ratio = negative/(positive+1)) %>%  
 ungroup()

## Warning in inner\_join(., get\_sentiments("nrc"), by = "word"): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 6 of `x` matches multiple rows in `y`.  
## ℹ Row 5657 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

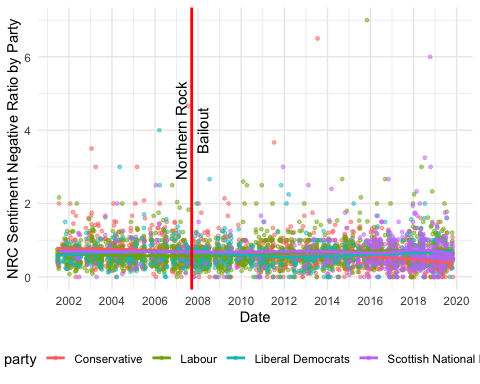
filtered\_nrc\_data\_party <-nrc\_data\_party %>%  
 group\_by(party) %>%  
 filter(n() > 100) %>% # party with more than 100 related words  
 ungroup()

Plotting a negative ratio plot of party. Points indicate the actual scores while smooth lines show the trends.

# A negative sentiment ratio plot of four parties  
filtered\_nrc\_data\_party %>%  
 ggplot(aes(date, ratio, color = party)) +  
 geom\_point(alpha=0.5, size = 1) +  
 geom\_smooth(method="loess", se = F, alpha=0.7, size = 1) +  
 scale\_x\_date(date\_breaks = "2 years", date\_labels = "%Y") +  
 labs(y = "NRC Sentiment Negative Ratio by Party", x = "Date") +  
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col="red", size = 1) +  
 annotate("text", x = as.Date("2007-09-24"), y = 4, label="Northern Rock\nBailout", angle=90, color = "black", size = 4) +  
 theme\_minimal() +  
 theme(legend.position = "bottom")

## 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.

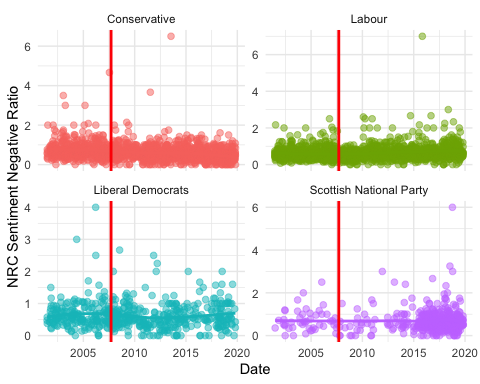
## `geom\_smooth()` using formula = 'y ~ x'

 Similar to the overall sentiment plot, there is no period when the proportion of negative words appears prominently.

Make the plot above readable, dividing plots by party

# Faceting by party  
filtered\_nrc\_data\_party %>%  
 ggplot(aes(x = date, y = ratio, color = party)) +  
 geom\_point(alpha = 0.5, size = 2) +  
 geom\_smooth(method = "loess", se = FALSE, alpha = 0.7, size = 1) +  
 labs(y = "NRC Sentiment Negative Ratio", x = "Date") +  
 facet\_wrap(~party, scales = "free\_y") +   
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col = "red", size = 1) +  
 theme\_minimal() +  
 theme(legend.position = "none")

## `geom\_smooth()` using formula = 'y ~ x'

 Similar to the overall sentiment plot, there is no period when the proportion of negative words appears prominently.

## 3) Sentiments counts by party

What is the most common sentiment? Summing up scores of sentiments by party to compare.

# Total sentiment word counts by party  
dominant\_senti\_nrc\_party <-filtered\_nrc\_data\_party %>%  
 group\_by(party) %>%  
 summarise(across(c(anger, anticipation, disgust, fear, joy, sadness, surprise, trust, negative, positive), sum, na.rm = T))

## Warning: There was 1 warning in `summarise()`.  
## ℹ In argument: `across(...)`.  
## ℹ In group 1: `party = "Conservative"`.  
## Caused by warning:  
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.  
## Supply arguments directly to `.fns` through an anonymous function instead.  
##   
## # Previously  
## across(a:b, mean, na.rm = TRUE)  
##   
## # Now  
## across(a:b, \(x) mean(x, na.rm = TRUE))

show(dominant\_senti\_nrc\_party)

## # A tibble: 4 × 11  
## party anger anticipation disgust fear joy sadness surprise trust negative  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Conser… 33439 52127 17861 56500 36474 32322 17703 99258 84545  
## 2 Labour 35105 53846 18420 57405 38456 34318 18408 97785 85101  
## 3 Libera… 5625 9303 2963 9620 5875 5778 3149 15899 14675  
## 4 Scotti… 4637 7398 2523 8180 4986 5388 2638 12649 12877  
## # ℹ 1 more variable: positive <dbl>

The large number of sentiment counts is generated by the Conservatives and Labor parties.

## 4) Sentiment changes by party

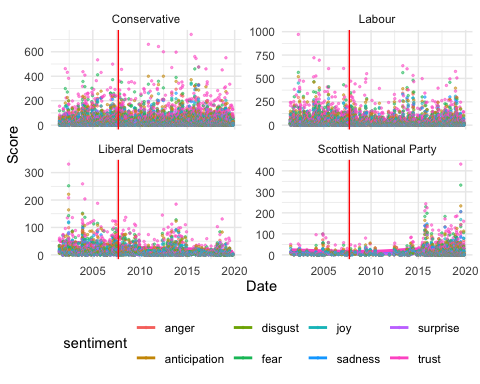
Beyond negative/positive ratio, focusing on specific sentiments trends by party.

# Combining all sentiments into one column  
long\_nrc\_data <- filtered\_nrc\_data\_party %>%  
 pivot\_longer(cols = c(anger, fear, trust, sadness, disgust, anticipation, surprise, joy), names\_to = "sentiment", values\_to = "score")

Faceting plots to compare the sentiment changes of each party

# Plotting specific sentiment changes by party  
long\_nrc\_data %>%  
 ggplot(aes(x = date, y = score, color = sentiment)) +  
 geom\_smooth(method = "loess", se = F, alpha = 1, size = 1) +  
 geom\_point(alpha = 0.5, size = 0.5) +   
 facet\_wrap(~ party, scales = "free\_y") +  
 labs(x = "Date", y = "Score") +  
 theme\_minimal() +  
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col = "red") +  
 theme\_minimal() +  
 theme(legend.position = "bottom")

## `geom\_smooth()` using formula = 'y ~ x'

 The smooth line in each plot shows a relatively steady and unchanged shape.

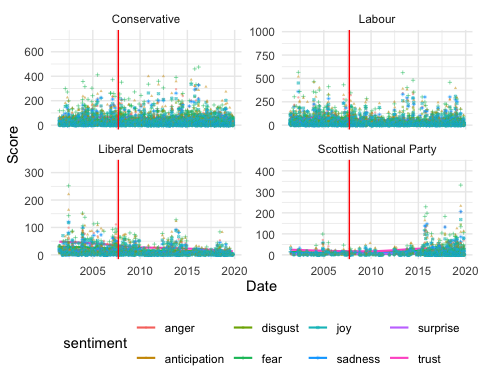
To make sentiment points recognizable, adjusting the shapes of points by sentiment and plotting

# Shaping sentiment points differently  
shape\_mapping <- c("anger" = 17, "disgust" = 18, "fear" = 19, "sadness" = 8, "trust" = 21, "anticipation" = 22, "surprise" = 23, "joy" = 24)  
  
long\_nrc\_data %>%  
 ggplot(aes(x = date, y = score, color = sentiment)) +  
 geom\_smooth(method = "loess", se = F, alpha = 0.75, size = 0.75) +   
 geom\_point(aes(shape = sentiment), alpha = 0.5, size = 0.5) +   
 facet\_wrap(~ party, scales = "free\_y", nrow = 2) +   
 labs(x = "Date", y = "Score") +  
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col = "red") +  
 theme\_minimal() +  
 theme(legend.position = "bottom")

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: The shape palette can deal with a maximum of 6 discrete values because more  
## than 6 becomes difficult to discriminate  
## ℹ you have requested 8 values. Consider specifying shapes manually if you need  
## that many have them.

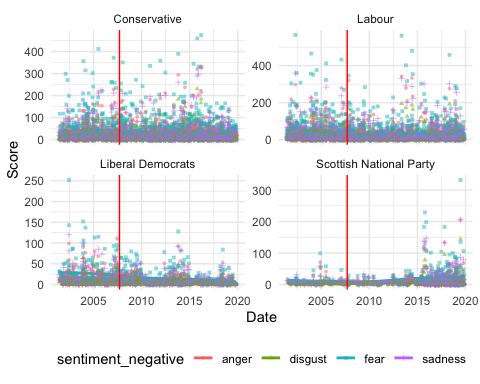
## Warning: Removed 8132 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



Selecting negative sentiments to confirm sentiment changes

# Specifying negative feelings only  
long\_nrc\_data\_neg <- filtered\_nrc\_data\_party %>%  
 pivot\_longer(cols = c(anger, fear, sadness, disgust), names\_to = "sentiment\_negative", values\_to = "score")  
  
# Plots of specific negative sentiments by party   
long\_nrc\_data\_neg %>%  
 ggplot(aes(x = date, y = score, color = sentiment\_negative)) +  
 geom\_smooth(method = "loess", se = F, alpha = 0.75, size = 1) +   
 geom\_point(aes(shape = sentiment\_negative), alpha = 0.5, size = 1) +   
 facet\_wrap(~ party, scales = "free\_y", nrow = 2) +   
 labs(x = "Date", y = "Score") +  
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col = "red") +  
 theme\_minimal() +  
 theme(legend.position = "bottom")

## `geom\_smooth()` using formula = 'y ~ x'

 Regarding negative sentiments, the Conservative party presents more diverse and fluctuated points than other parties. However, the general smooth lines stay steady.

## 5) Random sample sentiment

To compare with sentiment trends of immigration related text, make a random sample of 10000 and do sentiment analysis

# Random sampling  
data\_sample <- data %>%  
 sample\_n(10000)  
  
# tidy sample  
tidy\_samp <- data\_sample %>%  
 mutate(desc = tolower(text)) %>%  
 unnest\_tokens(word, desc) %>%  
 filter(str\_detect(word, "[a-z]")) %>%  
 filter(!word %in% stop\_words$word) %>%  
 arrange(date)  
  
tidy\_samp$order <- 1:nrow(tidy\_samp)  
  
# Applying NRC dictionary and calculating total negative ratio  
samp\_nrc\_data <- tidy\_samp %>%  
 inner\_join(get\_sentiments("nrc"), by = "word") %>%  
 count(date, sentiment) %>%  
 spread(key = sentiment, value = n, fill = 0) %>%  
 mutate(ratio = negative / (positive+1))

## Warning in inner\_join(., get\_sentiments("nrc"), by = "word"): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 2 of `x` matches multiple rows in `y`.  
## ℹ Row 5503 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

#Applying NRC dictionary and calculating negative ratio by party  
samp\_nrc\_party <- tidy\_samp %>%  
 inner\_join(get\_sentiments("nrc"), by = "word") %>%  
 group\_by(party, date) %>%  
 count(sentiment) %>%  
 spread(key = sentiment, value = n, fill = 0) %>%  
 mutate(ratio = negative / (positive+1)) %>%  
 ungroup()

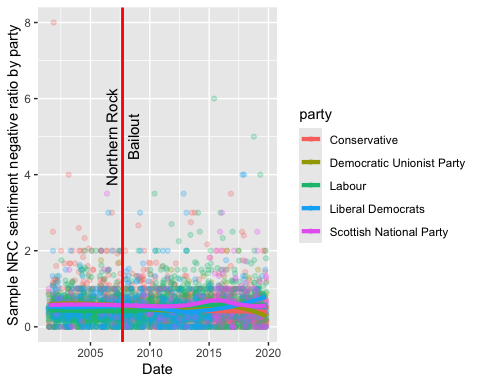
## Warning in inner\_join(., get\_sentiments("nrc"), by = "word"): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 2 of `x` matches multiple rows in `y`.  
## ℹ Row 5503 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

filtered\_samp\_nrc\_party <- samp\_nrc\_party %>%  
 group\_by(party) %>%  
 filter(n() > 100) %>%  
 ungroup()

A plot of sample text negative sentiment ratio by party

filtered\_samp\_nrc\_party %>%  
 ggplot(aes(date, ratio, color = party)) +  
 geom\_point(alpha=0.25) +  
 geom\_smooth(method="loess", se = F, alpha=0.25, size = 1.5) +  
 labs(y = "Sample NRC sentiment negative ratio by party", x = "Date") +  
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col = "red", size = 1) +   
 annotate("text", x = as.Date("2007-09-24"), y = 5, label="Northern Rock\nBailout", angle=90, color = "black", size = 4) +

theme\_minimal() +  
 theme(legend.position = "bottom")

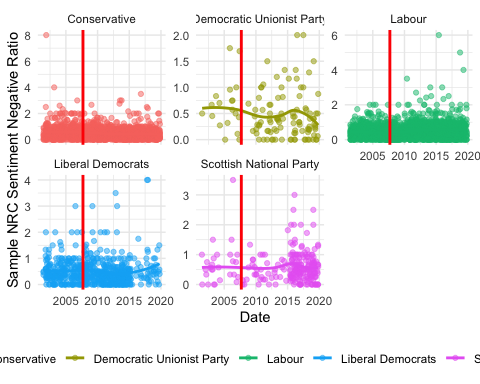


Negative sentiments of sample texts also show steady lines.

Dividing the sample sentiment plot by party

# Faceting plots by party  
filtered\_samp\_nrc\_party %>%  
 ggplot(aes(date, ratio, color = party)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = F, alpha = 0.7, size = 1) +  
 labs(y = "Sample NRC Sentiment Negative Ratio", x = "Date") +  
 facet\_wrap(~party, scales = "free\_y") +   
 geom\_vline(xintercept = as.numeric(as.Date("2007-09-14")), col = "red", size = 1) +  
 theme\_minimal() +  
 theme(legend.position = "bottom")

## `geom\_smooth()` using formula = 'y ~ x'

 Compared to the immigration related text sentiment plots, sample sentiment plots show no significant difference.

# 3. Word Embedding GloVe

# Set up  
library(text2vec)   
library(stringr)  
library(umap)  
library(ggrepel)

## 1) Training total immigration related texts

### (1) The GloVe model for total parliament speech

Training process

# choice parameters  
WINDOW\_SIZE <- 6 # context up to 6 words  
DIM <- 300 # the length of the word vector  
ITERS <- 100 # the maximum number of iterations  
COUNT\_MIN <- 10 # minimum count of words  
  
# shuffle text  
set.seed(42L) # for reproducibility  
glove\_text <- sample(tidy\_data\_notoken$desc)  
  
# create vocab  
tokens <- space\_tokenizer(glove\_text) # tokenizing  
it <- itoken(tokens, progressbar = FALSE) # create the vocabulary object  
vocab <- create\_vocabulary(it) # create a vocabulary  
vocab\_pruned <- prune\_vocabulary(vocab, term\_count\_min = COUNT\_MIN) # keep only words that meet count threshold  
  
#vectorize and create term co-occurrence matrix  
vectorizer <- vocab\_vectorizer(vocab\_pruned)  
tcm <- create\_tcm(it, vectorizer, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric",   
 weights = rep(1, WINDOW\_SIZE))

Setting model parameters

#set model parameters  
glove <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
  
# fit model  
word\_vectors\_main <- glove$fit\_transform(tcm, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
# get output   
word\_vectors\_context <- glove$components  
glove\_embedding <- word\_vectors\_main + t(word\_vectors\_context) # word vectors. combine main and context word vectors  
  
#save   
saveRDS(glove\_embedding, file = "local\_glove.rds")

Modelling takes time. So I will use the prepared model made by same process for knitting.

url <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove.rds?raw=true"  
glove\_embedding <- readRDS(url(url, method = "libcurl"))

Visulalization of GloVe word embedding model by using umap.

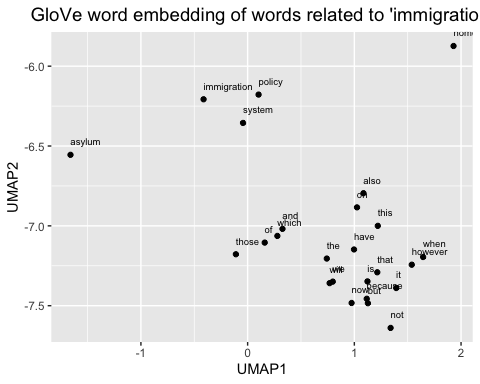
# GloVe dimension reduction (two dimension)  
glove\_umap <- umap(glove\_embedding, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread=2)  
  
# Put results in a dataframe for ggplot  
df\_glove\_umap <- as.data.frame(glove\_umap[["layout"]])  
  
# Add the labels of the words to the dataframe  
df\_glove\_umap$word <- rownames(df\_glove\_umap)  
colnames(df\_glove\_umap) <- c("UMAP1", "UMAP2", "word")

Plot the total word embedding of ‘immigration’ with the GloVe model generated above.

word <- glove\_embedding["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding, y = word, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words <- df\_glove\_umap %>%   
 inner\_join(y=select, by= "word")

The ggplot visual for GloVe regarding total immigration related texts. Total parliament speech word embedding of words related to *immigration*

ggplot(selected\_words, aes(x = UMAP1, y = UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(UMAP1, UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "GloVe word embedding of words related to 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))

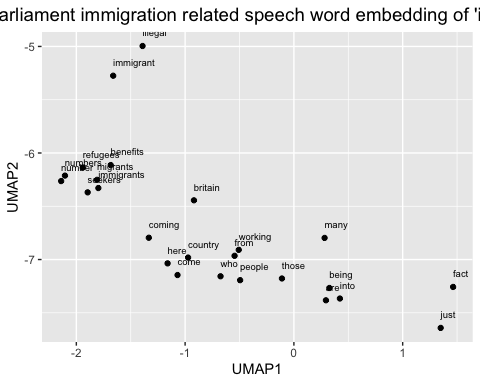


I did it for confirming the modeling process, but we might need to use this map as a comparison plot.

Total parliament speech word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (Case2: immigrants)  
word\_2 <- glove\_embedding["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding, y = word\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_2 <- df\_glove\_umap %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_2, aes(x = UMAP1, y = UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(UMAP1, UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Total parliament immigration related speech word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))

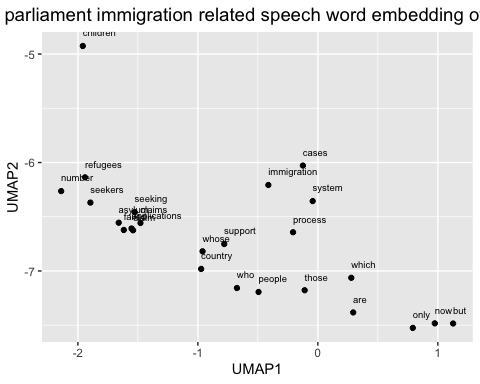


This can be also used for comparison. Also, word embedding maps tell us the general relationships between words, especially those we are interested in.

For the last pilot modelling map, there is a total parliament immigration related speech word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (asylum)  
word\_4 <- glove\_embedding["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding, y = word\_4, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_4 <- df\_glove\_umap %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_4, aes(x = UMAP1, y = UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(UMAP1, UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Total parliament immigration related speech word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



### (2) The Glove model for total Parliament speech before the 2008 economic crisis

# filter data by date. Now have pre/post economic crisis text data related to immigration.  
  
data\_pre <- filter(tidy\_data\_notoken, date < as.Date("2007-09-24"))  
data\_post <- filter(tidy\_data\_notoken, date >= as.Date("2007-09-24"))

# repeat the same modeling process  
set.seed(42L)  
glove\_text\_pre <- sample(data\_pre$desc)  
  
tokens\_pre <- space\_tokenizer(glove\_text\_pre)  
it\_pre <- itoken(tokens\_pre, progressbar = FALSE)  
vocab\_pre <- create\_vocabulary(it\_pre)  
vocab\_pruned\_pre <- prune\_vocabulary(vocab\_pre, term\_count\_min = COUNT\_MIN)  
  
vectorizer\_pre <- vocab\_vectorizer(vocab\_pruned\_pre)  
tcm\_pre <- create\_tcm(it\_pre, vectorizer\_pre, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric", weights = rep(1, WINDOW\_SIZE))  
  
glove\_pre <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
word\_vectors\_main\_pre <- glove\_pre$fit\_transform(tcm\_pre, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
word\_vectors\_context\_pre <- glove\_pre$components  
glove\_embedding\_pre <- word\_vectors\_main\_pre + t(word\_vectors\_context\_pre)  
  
saveRDS(glove\_embedding\_pre, file = "local\_glove\_pre.rds")

Modelling takes time. So I will use the prepared model made by same process for knitting.

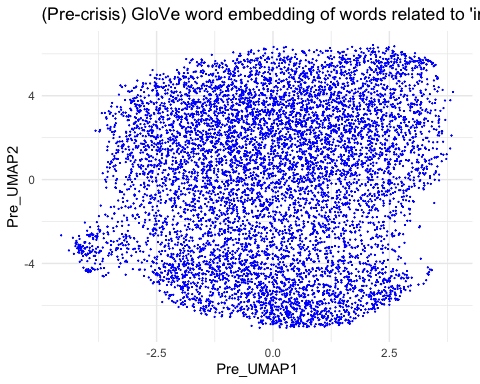
url\_pre <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove\_pre.rds?raw=true"  
glove\_embedding\_pre <- readRDS(url(url\_pre, method = "libcurl"))

Pre-crisis GloVe word embedding model of total parliament texts by two dimensional umap

# Plotting the whole word embedding of pre-crisis immigration related text  
umap\_pre <- umap(glove\_embedding\_pre, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread = 2)  
  
df\_umap\_pre <- as.data.frame(umap\_pre[["layout"]])  
df\_umap\_pre$word <- rownames(df\_umap\_pre)  
colnames(df\_umap\_pre) <- c("Pre\_UMAP1", "Pre\_UMAP2", "word")

Let’s see the whole word embedding map of two dimensions

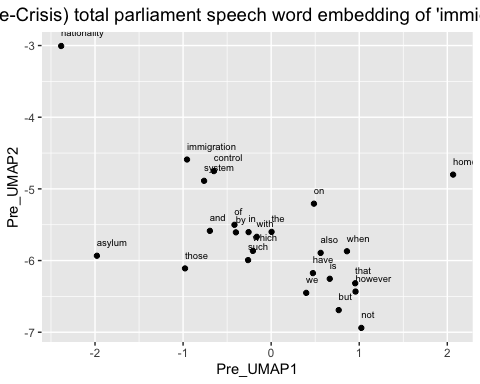
ggplot(df\_umap\_pre) +  
 geom\_point(aes(x = Pre\_UMAP1, y = Pre\_UMAP2), color = 'blue', size = 0.05) +  
 labs(title = "(Pre-crisis) GloVe word embedding of words related to 'immigration'") +  
 theme\_minimal()



Specify the map. Which words are close to the word ‘immigration’ in the total parliament texts? Pre-crisis total parliament speech word embedding of words related to *immigration*

# Plot the word embedding of words that are related for the GloVe model (Case1: immigration)  
word\_pre\_1 <- glove\_embedding\_pre["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_pre, y = word\_pre\_1, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_pre\_1 <- df\_umap\_pre %>%   
 inner\_join(y=select, by= "word")

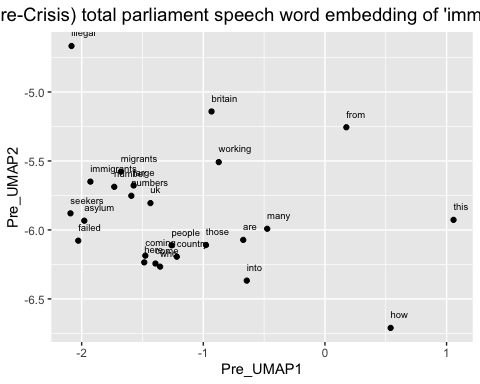
#The ggplot visual for GloVe  
ggplot(selected\_words\_pre\_1, aes(x = Pre\_UMAP1, y = Pre\_UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_UMAP1, Pre\_UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "(Pre-Crisis) total parliament speech word embedding of 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis total parliament speech word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (Case2: immigrants)  
word\_pre\_2 <- glove\_embedding\_pre["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_pre, y = word\_pre\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_pre\_2 <- df\_umap\_pre %>%   
 inner\_join(y=select, by= "word")

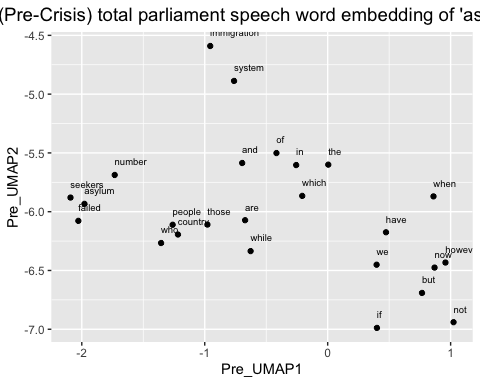
#The ggplot visual for GloVe  
ggplot(selected\_words\_pre\_2, aes(x = Pre\_UMAP1, y = Pre\_UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_UMAP1, Pre\_UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "(Pre-Crisis) total parliament speech word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis total parliament speech word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (asylum)  
word\_pre\_3 <- glove\_embedding\_pre["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_pre, y = word\_pre\_3, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_pre\_3 <- df\_umap\_pre %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_pre\_3, aes(x = Pre\_UMAP1, y = Pre\_UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_UMAP1, Pre\_UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "(Pre-Crisis) total parliament speech word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Now we can check the words relationships before the 2008 financail crisis.

### (3) The Glove model for total Parliament speech after the 2008 financial crisis

Again, same modeling process

set.seed(42L)  
glove\_text\_post <- sample(data\_post$desc)  
  
tokens\_post <- space\_tokenizer(glove\_text\_post)  
it\_post <- itoken(tokens\_post, progressbar = FALSE)  
vocab\_post <- create\_vocabulary(it\_post)  
vocab\_pruned\_post <- prune\_vocabulary(vocab\_post, term\_count\_min = COUNT\_MIN)  
  
vectorizer\_post <- vocab\_vectorizer(vocab\_pruned\_post)  
tcm\_post <- create\_tcm(it\_post, vectorizer\_post, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric", weights = rep(1, WINDOW\_SIZE))  
  
glove\_post <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
word\_vectors\_main\_post <- glove\_post$fit\_transform(tcm\_post, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
word\_vectors\_context\_post <- glove\_post$components  
glove\_embedding\_post <- word\_vectors\_main\_post + t(word\_vectors\_context\_post)  
  
saveRDS(glove\_embedding\_post, file = "local\_glove\_post.rds")

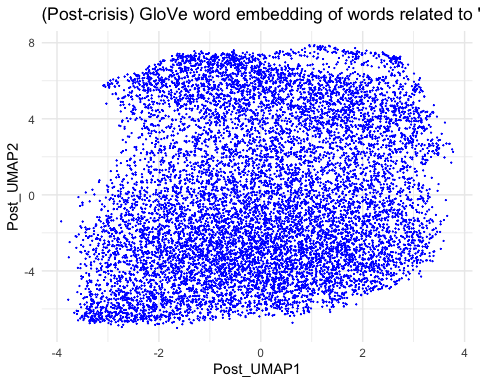
Modelling takes time. So I will use the prepared model made by same process for knitting.

url\_post <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove\_post.rds?raw=true"  
glove\_embedding\_post <- readRDS(url(url\_post, method = "libcurl"))

Pre-crisis total GloVe word embedding two dimensional umap

# Plotting the whole word embedding of pre-crisis immigration related text  
umap\_post <- umap(glove\_embedding\_post, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread = 2)  
  
df\_umap\_post <- as.data.frame(umap\_post[["layout"]])  
df\_umap\_post$word <- rownames(df\_umap\_post)  
colnames(df\_umap\_post) <- c("Post\_UMAP1", "Post\_UMAP2", "word")

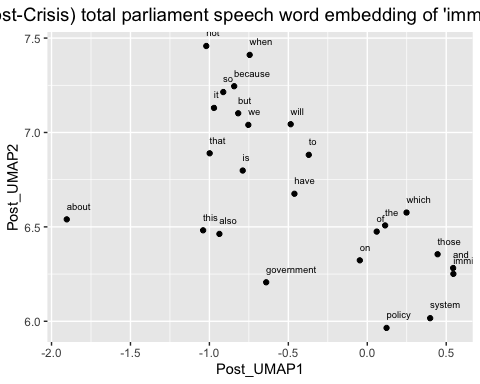
ggplot(df\_umap\_post) +  
 geom\_point(aes(x = Post\_UMAP1, y = Post\_UMAP2), color = 'blue', size = 0.05) +  
 labs(title = "(Post-crisis) GloVe word embedding of words related to 'immigration'") +  
 theme\_minimal()



Pre-crisis total parliament speech word embedding of words related to *immigration*

# Plot the word embedding of words that are related for the GloVe model (Case1: immigration)  
word\_post\_1 <- glove\_embedding\_post["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_post, y = word\_post\_1, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_post\_1 <- df\_umap\_post %>%   
 inner\_join(y=select, by= "word")

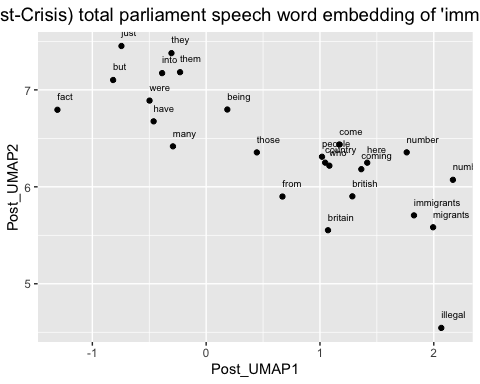
#The ggplot visual for GloVe  
ggplot(selected\_words\_post\_1, aes(x = Post\_UMAP1, y = Post\_UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_UMAP1, Post\_UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "(Post-Crisis) total parliament speech word embedding of 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis total parliament speech word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (Case2: immigrants)  
word\_post\_2 <- glove\_embedding\_post["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_post, y = word\_post\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_post\_2 <- df\_umap\_post %>%   
 inner\_join(y=select, by= "word")

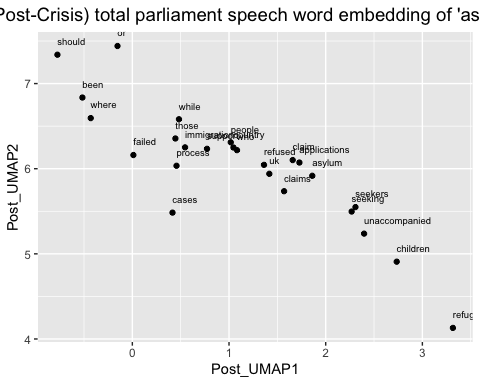
#The ggplot visual for GloVe  
ggplot(selected\_words\_post\_2, aes(x = Post\_UMAP1, y = Post\_UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_UMAP1, Post\_UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "(Post-Crisis) total parliament speech word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Post-crisis total parliament speech word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (asylum)  
word\_post\_4 <- glove\_embedding\_post["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_post, y = word\_post\_4, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_post\_4 <- df\_umap\_post %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_post\_4, aes(x = Post\_UMAP1, y = Post\_UMAP2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_UMAP1, Post\_UMAP2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "(Post-Crisis) total parliament speech word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))

 We can now compare the two periods with three different embedding maps.

# 2) Training by party and pre/post economic crisis

To make four different GloVe embedding models, dividing data by party and date

library(RcppParallel)  
  
# filter data by party and date. Now have pre/post economic crisis text data of Conservative and Labour party   
data\_conser\_pre <- filter(tidy\_data\_notoken, party == "Conservative" & date < as.Date("2007-09-24"))  
data\_labour\_pre <- filter(tidy\_data\_notoken, party == "Labour" & date < as.Date("2007-09-24"))  
  
data\_conser\_post <- filter(tidy\_data\_notoken, party == "Conservative" & date >= as.Date("2007-09-24"))  
data\_labour\_post <- filter(tidy\_data\_notoken, party == "Labour" & date >= as.Date("2007-09-24"))

### (1) The GloVe model for Conservative party before the 2008 economic crisis

Training process

set.seed(42L)  
glove\_text\_conser\_pre <- sample(data\_conser\_pre$desc)  
  
tokens\_conser\_pre <- space\_tokenizer(glove\_text\_conser\_pre)  
it\_conser\_pre <- itoken(tokens\_conser\_pre, progressbar = FALSE)  
vocab\_conser\_pre <- create\_vocabulary(it\_conser\_pre)  
vocab\_pruned\_conser\_pre <- prune\_vocabulary(vocab\_conser\_pre, term\_count\_min = COUNT\_MIN)  
  
vectorizer\_conser\_pre <- vocab\_vectorizer(vocab\_pruned\_conser\_pre)  
tcm\_conser\_pre <- create\_tcm(it\_conser\_pre, vectorizer\_conser\_pre, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric", weights = rep(1, WINDOW\_SIZE))  
  
glove\_conser\_pre <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
word\_vectors\_main\_conser\_pre <- glove\_conser\_pre$fit\_transform(tcm\_conser\_pre, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
word\_vectors\_context\_conser\_pre <- glove\_conser\_pre$components  
glove\_embedding\_conser\_pre <- word\_vectors\_main\_conser\_pre + t(word\_vectors\_context\_conser\_pre)  
  
saveRDS(glove\_embedding\_conser\_pre, file = "local\_glove\_conser\_pre.rds")

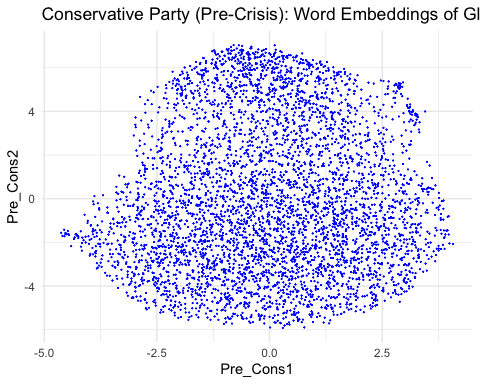
Modelling takes time. So I will use the prepared model made by same process for knitting.

url\_conser\_pre <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove\_conser\_pre.rds?raw=true"  
glove\_embedding\_conser\_pre <- readRDS(url(url\_conser\_pre, method = "libcurl"))

Total GloVe word embedding two dimensional umap

# Plotting the whole word embeddings of pre-crisis Conservative party immigration related text  
umap\_conser\_pre <- umap(glove\_embedding\_conser\_pre, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread = 2)  
  
df\_umap\_conser\_pre <- as.data.frame(umap\_conser\_pre[["layout"]])  
df\_umap\_conser\_pre$word <- rownames(df\_umap\_conser\_pre)  
colnames(df\_umap\_conser\_pre) <- c("Pre\_Cons1", "Pre\_Cons2", "word")

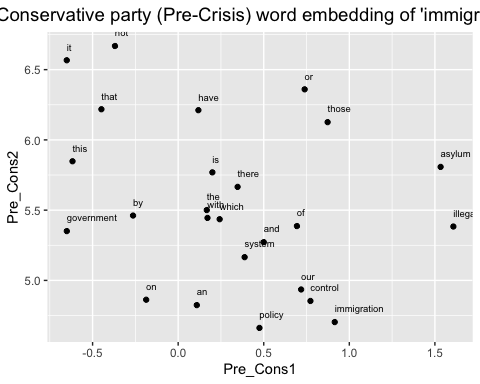
ggplot(df\_umap\_conser\_pre) +  
 geom\_point(aes(x = Pre\_Cons1, y = Pre\_Cons2), color = 'blue', size = 0.05) +  
 labs(title = "Conservative Party (Pre-Crisis): Word Embeddings of GloVe and UMAP") +  
 theme\_minimal()



Pre-crisis Conservative party word embedding of words related to *immigration*

# Plot the word embedding of words that are related for the GloVe model (Case1: immigration)  
word\_conser\_pre\_1 <- glove\_embedding\_conser\_pre["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_conser\_pre, y = word\_conser\_pre\_1, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_conser\_pre\_1 <- df\_umap\_conser\_pre %>%   
 inner\_join(y=select, by= "word")

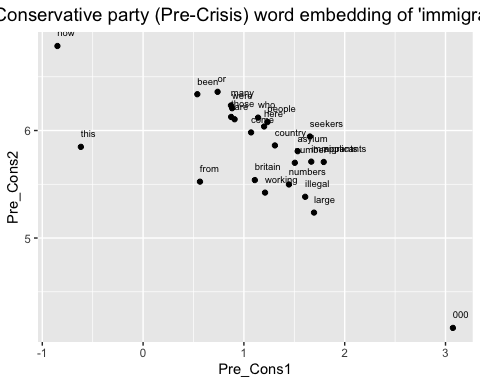
#The ggplot visual for GloVe  
ggplot(selected\_words\_conser\_pre\_1, aes(x = Pre\_Cons1, y = Pre\_Cons2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_Cons1, Pre\_Cons2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Conservative party (Pre-Crisis) word embedding of 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis Conservative party word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (Case2: immigrants)  
word\_conser\_pre\_2 <- glove\_embedding\_conser\_pre["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_conser\_pre, y = word\_conser\_pre\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_conser\_pre\_2 <- df\_umap\_conser\_pre %>%   
 inner\_join(y=select, by= "word")

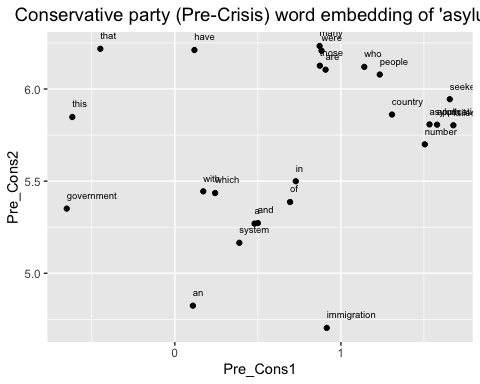
#The ggplot visual for GloVe  
ggplot(selected\_words\_conser\_pre\_2, aes(x = Pre\_Cons1, y = Pre\_Cons2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_Cons1, Pre\_Cons2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Conservative party (Pre-Crisis) word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis Conservative party word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (case3: asylum)  
word\_conser\_pre\_3 <- glove\_embedding\_conser\_pre["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_conser\_pre, y = word\_conser\_pre\_3, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_conser\_pre\_3 <- df\_umap\_conser\_pre %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_conser\_pre\_3, aes(x = Pre\_Cons1, y = Pre\_Cons2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_Cons1, Pre\_Cons2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Conservative party (Pre-Crisis) word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



### (2) The GloVe model for Conservative party after the 2008 economic crisis

Training process

set.seed(42L)  
glove\_text\_conser\_post <- sample(data\_conser\_post$desc)  
  
tokens\_conser\_post <- space\_tokenizer(glove\_text\_conser\_post)  
it\_conser\_post <- itoken(tokens\_conser\_post, progressbar = FALSE)  
vocab\_conser\_post <- create\_vocabulary(it\_conser\_post)  
vocab\_pruned\_conser\_post <- prune\_vocabulary(vocab\_conser\_post, term\_count\_min = COUNT\_MIN)  
  
vectorizer\_conser\_post <- vocab\_vectorizer(vocab\_pruned\_conser\_post)  
tcm\_conser\_post <- create\_tcm(it\_conser\_post, vectorizer\_conser\_post, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric", weights = rep(1, WINDOW\_SIZE))  
  
glove\_conser\_post <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
word\_vectors\_main\_conser\_post <- glove\_conser\_post$fit\_transform(tcm\_conser\_post, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
word\_vectors\_context\_conser\_post <- glove\_conser\_post$components  
glove\_embedding\_conser\_post <- word\_vectors\_main\_conser\_post + t(word\_vectors\_context\_conser\_post)  
  
saveRDS(glove\_embedding\_conser\_post, file = "local\_glove\_conser\_post.rds")

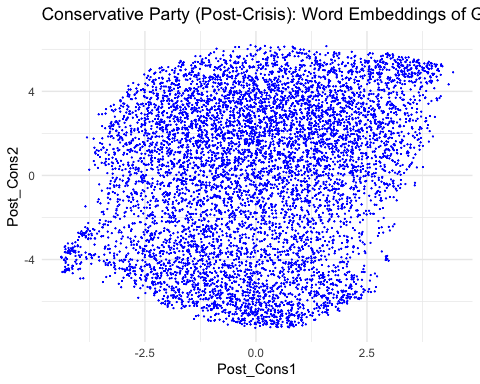
Modelling takes time. So I will use the prepared model made by same process for knitting.

url\_conser\_post <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove\_conser\_post.rds?raw=true"  
glove\_embedding\_conser\_post <- readRDS(url(url\_conser\_post, method = "libcurl"))

Total GloVe word embedding two dimensional umap

# Plotting the whole word embeddings of post-crisis Conservative party immigration related text  
umap\_conser\_post <- umap(glove\_embedding\_conser\_post, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread = 2)  
  
df\_umap\_conser\_post <- as.data.frame(umap\_conser\_post[["layout"]])  
df\_umap\_conser\_post$word <- rownames(df\_umap\_conser\_post)  
colnames(df\_umap\_conser\_post) <- c("Post\_Cons1", "Post\_Cons2", "word")

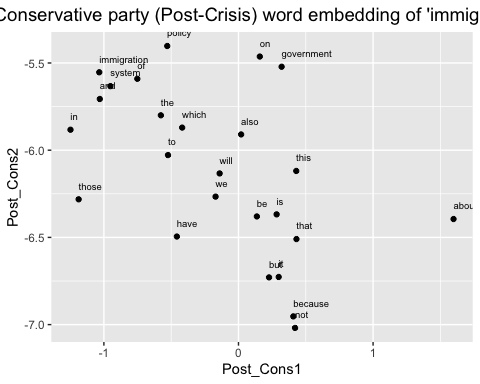
ggplot(df\_umap\_conser\_post) +  
 geom\_point(aes(x = Post\_Cons1, y = Post\_Cons2), color = 'blue', size = 0.05) +  
 labs(title = "Conservative Party (Post-Crisis): Word Embeddings of GloVe and UMAP") +  
 theme\_minimal()



Post-crisis Conservative party word embedding of words related to *immigration*

# Plot the word embedding of words that are related for the GloVe model (Case1: immigration)  
word\_conser\_post\_1 <- glove\_embedding\_conser\_post["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_conser\_post, y = word\_conser\_post\_1, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_conser\_post\_1 <- df\_umap\_conser\_post %>%   
 inner\_join(y=select, by= "word")

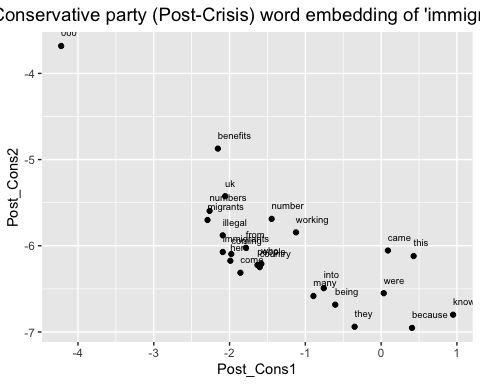
#The ggplot visual for GloVe  
ggplot(selected\_words\_conser\_post\_1, aes(x = Post\_Cons1, y = Post\_Cons2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_Cons1, Post\_Cons2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Conservative party (Post-Crisis) word embedding of 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Post-crisis Conservative party word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (case2: immigrants)  
word\_conser\_post\_2 <- glove\_embedding\_conser\_post["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_conser\_post, y = word\_conser\_post\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_conser\_post\_2 <- df\_umap\_conser\_post %>%   
 inner\_join(y=select, by= "word")

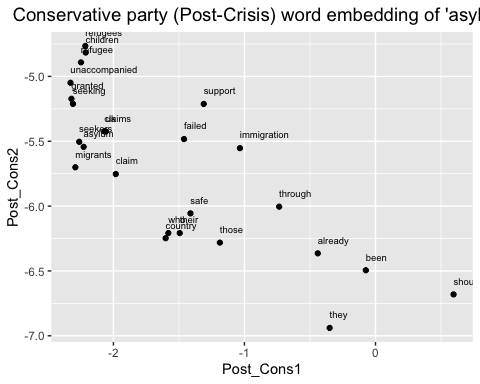
#The ggplot visual for GloVe  
ggplot(selected\_words\_conser\_post\_2, aes(x = Post\_Cons1, y = Post\_Cons2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_Cons1, Post\_Cons2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Conservative party (Post-Crisis) word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Post-crisis Conservative party word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (Case3: asylum)  
word\_conser\_post\_4 <- glove\_embedding\_conser\_post["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_conser\_post, y = word\_conser\_post\_4, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_conser\_post\_4 <- df\_umap\_conser\_post %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_conser\_post\_4, aes(x = Post\_Cons1, y = Post\_Cons2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_Cons1, Post\_Cons2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Conservative party (Post-Crisis) word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



### (3) The GloVe model for Labour party before the 2008 economic crisis

Training process

set.seed(42L)  
glove\_text\_labour\_pre <- sample(data\_labour\_pre$desc)  
  
tokens\_labour\_pre <- space\_tokenizer(glove\_text\_labour\_pre)  
it\_labour\_pre <- itoken(tokens\_labour\_pre, progressbar = FALSE)  
vocab\_labour\_pre <- create\_vocabulary(it\_labour\_pre)  
vocab\_pruned\_labour\_pre <- prune\_vocabulary(vocab\_labour\_pre, term\_count\_min = COUNT\_MIN)  
  
vectorizer\_labour\_pre <- vocab\_vectorizer(vocab\_pruned\_labour\_pre)  
tcm\_labour\_pre <- create\_tcm(it\_labour\_pre, vectorizer\_labour\_pre, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric", weights = rep(1, WINDOW\_SIZE))  
  
glove\_labour\_pre <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
word\_vectors\_main\_labour\_pre <- glove\_labour\_pre$fit\_transform(tcm\_labour\_pre, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
word\_vectors\_context\_labour\_pre <- glove\_labour\_pre$components  
glove\_embedding\_labour\_pre <- word\_vectors\_main\_labour\_pre + t(word\_vectors\_context\_labour\_pre)  
  
saveRDS(glove\_embedding\_labour\_pre, file = "local\_glove\_labour\_pre.rds")

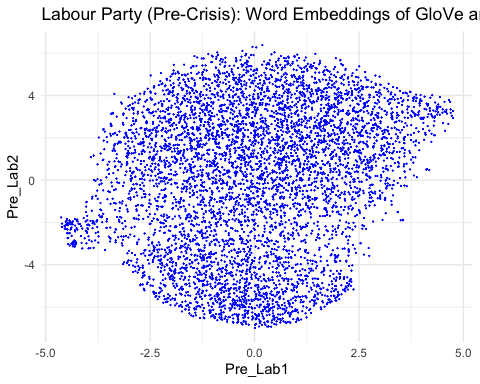
Modelling takes time. So I will use the prepared model made by same process for knitting.

url\_labour\_pre <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove\_labour\_pre.rds?raw=true"  
glove\_embedding\_labour\_pre <- readRDS(url(url\_labour\_pre, method = "libcurl"))

Total GloVe word embedding two dimensional umap

# Plotting the whole word embeddings of pre-crisis Labour party immigration related text  
umap\_labour\_pre <- umap(glove\_embedding\_labour\_pre, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread = 2)  
  
df\_umap\_labour\_pre <- as.data.frame(umap\_labour\_pre[["layout"]])  
df\_umap\_labour\_pre$word <- rownames(df\_umap\_labour\_pre)  
colnames(df\_umap\_labour\_pre) <- c("Pre\_Lab1", "Pre\_Lab2", "word")

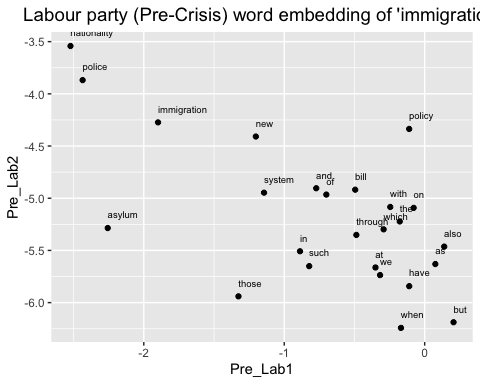
ggplot(df\_umap\_labour\_pre) +  
 geom\_point(aes(x = Pre\_Lab1, y = Pre\_Lab2), color = 'blue', size = 0.05) +  
 labs(title = "Labour Party (Pre-Crisis): Word Embeddings of GloVe and UMAP") +  
 theme\_minimal()



Pre-crisis Labour party word embedding of words related to *immigration*

# Plot the word embedding of words that are related for the GloVe model (case1: immigration)  
word\_labour\_pre\_1 <- glove\_embedding\_labour\_pre["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_labour\_pre, y = word\_labour\_pre\_1, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_labour\_pre\_1 <- df\_umap\_labour\_pre %>%   
 inner\_join(y=select, by= "word")

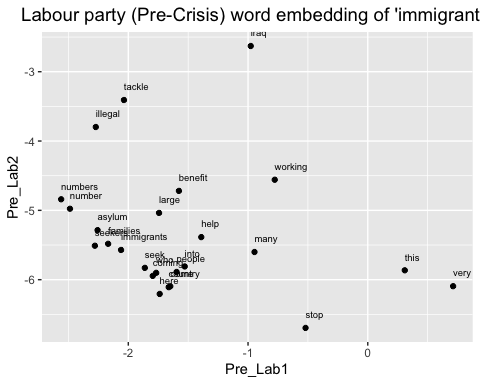
#The ggplot visual for GloVe  
ggplot(selected\_words\_labour\_pre\_1, aes(x = Pre\_Lab1, y = Pre\_Lab2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_Lab1, Pre\_Lab2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Labour party (Pre-Crisis) word embedding of 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis Labour party word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (Case2: immigrants)  
word\_labour\_pre\_2 <- glove\_embedding\_labour\_pre["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_labour\_pre, y = word\_labour\_pre\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_labour\_pre\_2 <- df\_umap\_labour\_pre %>%   
 inner\_join(y=select, by= "word")

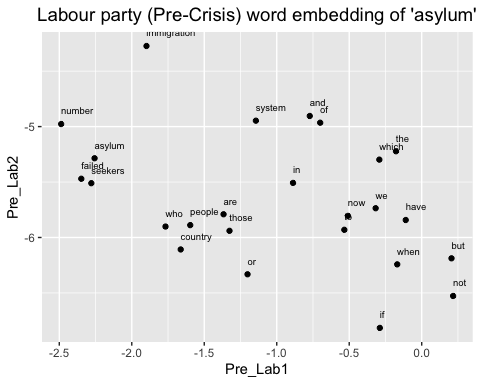
#The ggplot visual for GloVe  
ggplot(selected\_words\_labour\_pre\_2, aes(x = Pre\_Lab1, y = Pre\_Lab2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_Lab1, Pre\_Lab2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Labour party (Pre-Crisis) word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis Labour party word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (case3: asylum)  
word\_labour\_pre\_3 <- glove\_embedding\_labour\_pre["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_labour\_pre, y = word\_labour\_pre\_3, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_labour\_pre\_3 <- df\_umap\_labour\_pre %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_labour\_pre\_3, aes(x = Pre\_Lab1, y = Pre\_Lab2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Pre\_Lab1, Pre\_Lab2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Labour party (Pre-Crisis) word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



### (4) The GloVe model for Labour party after the 2008 economic crisis

Training process

set.seed(42L)  
glove\_text\_labour\_post <- sample(data\_labour\_post$desc)  
  
tokens\_labour\_post <- space\_tokenizer(glove\_text\_labour\_post)  
it\_labour\_post <- itoken(tokens\_labour\_post, progressbar = FALSE)  
vocab\_labour\_post <- create\_vocabulary(it\_labour\_post)  
vocab\_pruned\_labour\_post <- prune\_vocabulary(vocab\_labour\_post, term\_count\_min = COUNT\_MIN)  
  
vectorizer\_labour\_post <- vocab\_vectorizer(vocab\_pruned\_labour\_post)  
tcm\_labour\_post <- create\_tcm(it\_labour\_post, vectorizer\_labour\_post, skip\_grams\_window = WINDOW\_SIZE, skip\_grams\_window\_context = "symmetric", weights = rep(1, WINDOW\_SIZE))  
  
glove\_labour\_post <- GlobalVectors$new(rank = DIM, x\_max = 100, learning\_rate = 0.05)  
word\_vectors\_main\_labour\_post <- glove\_labour\_post$fit\_transform(tcm\_labour\_post, n\_iter = ITERS, convergence\_tol = 0.001, n\_threads = RcppParallel::defaultNumThreads())  
  
word\_vectors\_context\_labour\_post <- glove\_labour\_post$components  
glove\_embedding\_labour\_post <- word\_vectors\_main\_labour\_post + t(word\_vectors\_context\_labour\_post)  
  
saveRDS(glove\_embedding\_labour\_post, file = "local\_glove\_labour\_post.rds")

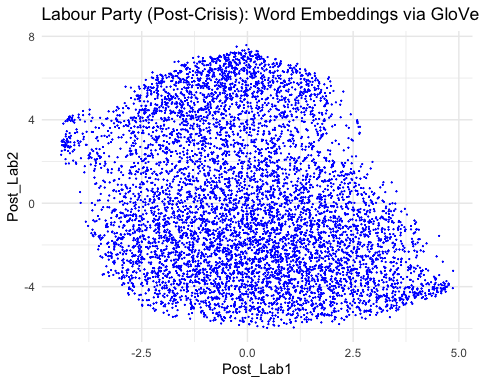
Modelling takes time. So I will use the prepared model made by same process for knitting.

url\_labour\_post <- "https://github.com/RiverKim-garam/CTA24-Final-assessment/blob/main/local\_glove\_labour\_post.rds?raw=true"  
glove\_embedding\_labour\_post <- readRDS(url(url\_labour\_post, method = "libcurl"))

Total GloVe word embedding two dimensional umap

# Plotting the whole word embeddings of post-crisis Labour party immigration related text  
umap\_labour\_post <- umap(glove\_embedding\_labour\_post, n\_components = 2, metric = "cosine", n\_neighbors = 25, min\_dist = 0.1, spread = 2)  
  
df\_umap\_labour\_post <- as.data.frame(umap\_labour\_post[["layout"]])  
df\_umap\_labour\_post$word <- rownames(df\_umap\_labour\_post)  
colnames(df\_umap\_labour\_post) <- c("Post\_Lab1", "Post\_Lab2", "word")

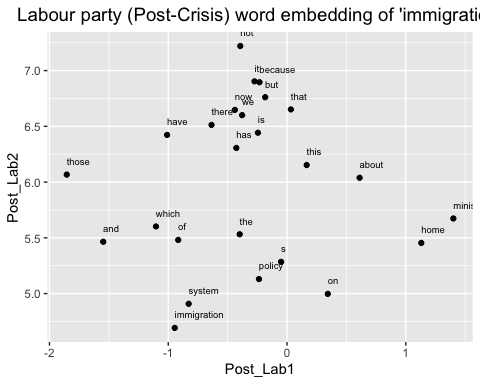
ggplot(df\_umap\_labour\_post) +  
 geom\_point(aes(x = Post\_Lab1, y = Post\_Lab2), color = 'blue', size = 0.05) +  
 labs(title = "Labour Party (Post-Crisis): Word Embeddings via GloVe and UMAP") +  
 theme\_minimal()



Post-crisis Labour party word embedding of words related to *immigration*

# Plot the word embedding of words that are related for the GloVe model (case1: immigration)  
word\_labour\_post\_1 <- glove\_embedding\_labour\_post["immigration",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_labour\_post, y = word\_labour\_post\_1, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_labour\_post\_1 <- df\_umap\_labour\_post %>%   
 inner\_join(y=select, by= "word")

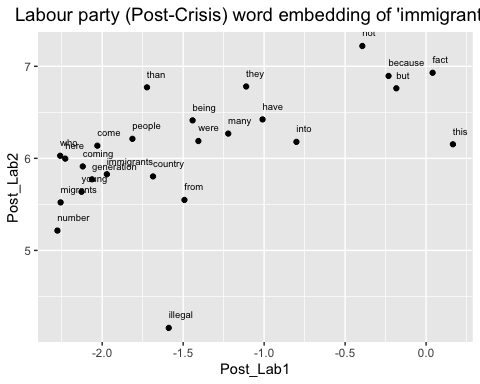
#The ggplot visual for GloVe  
ggplot(selected\_words\_labour\_post\_1, aes(x = Post\_Lab1, y = Post\_Lab2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_Lab1, Post\_Lab2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Labour party (Post-Crisis) word embedding of 'immigration'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis Labour party word embedding of words related to *immigrants*

# Plot the word embedding of words that are related for the GloVe model (case2: immigrants)  
word\_labour\_post\_2 <- glove\_embedding\_labour\_post["immigrants",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_labour\_post, y = word\_labour\_post\_2, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_labour\_post\_2 <- df\_umap\_labour\_post %>%   
 inner\_join(y=select, by= "word")

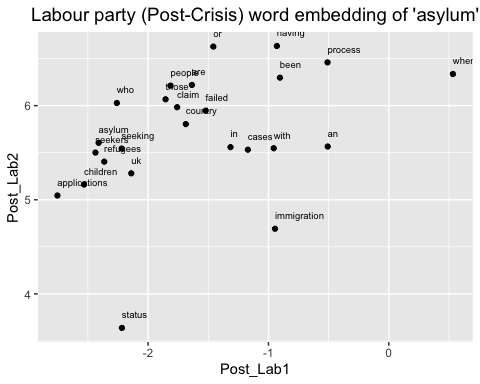
#The ggplot visual for GloVe  
ggplot(selected\_words\_labour\_post\_2, aes(x = Post\_Lab1, y = Post\_Lab2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_Lab1, Post\_Lab2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Labour party (Post-Crisis) word embedding of 'immigrants'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Pre-crisis Labour party word embedding of words related to *asylum*

# Plot the word embedding of words that are related for the GloVe model (case3: asylum)  
word\_labour\_post\_3 <- glove\_embedding\_labour\_post["asylum",, drop = FALSE]  
cos\_sim = sim2(x = glove\_embedding\_labour\_post, y = word\_labour\_post\_3, method = "cosine", norm = "l2")  
select <- data.frame(rownames(as.data.frame(head(sort(cos\_sim[,1], decreasing = TRUE), 25))))  
colnames(select) <- "word"  
selected\_words\_labour\_post\_3 <- df\_umap\_labour\_post %>%   
 inner\_join(y=select, by= "word")

#The ggplot visual for GloVe  
ggplot(selected\_words\_labour\_post\_3, aes(x = Post\_Lab1, y = Post\_Lab2)) +   
 geom\_point(show.legend = FALSE) +   
 geom\_text(aes(Post\_Lab1, Post\_Lab2, label = word), show.legend = FALSE, size = 2.5, vjust=-1.5, hjust=0) +  
 labs(title = "Labour party (Post-Crisis) word embedding of 'asylum'") +  
 theme(plot.title = element\_text(hjust = .5, size = 14))



Now we made three different specific words (immigration, immigrants, asylum) GloVE embedding umap by party and the period. By comparing the connections of words, we can estimate the contexts how the words were used.

# Instead of LaTex, use TinyTeX to knit the mark down result into pdf  
library(tinytex)