Text Mining Analysis on Artificial Intelligence Articles Surrounding Human Rights Policy and Performance

June 11, 2024
Anugya Mishra

Introduction

Artificial Intelligence (AI) has been a hot topic for quite a while now. With the introduction of ChatGPT by OpenAI at the end of 2022, consumer awareness about AI technology increased even more in the past year, and generative AI tools are being widely utilized in today's world for a variety of purposes. Similarly, the success of ChatGPT prompted companies across all industries to increase their efforts in harnessing AI technology, leading to rapid evolution in AI capabilities.

However, with AI technology gaining more attention, people are not only realizing the numerous opportunities but also the challenges and ethical concerns surrounding this widespread AI adoption. Various concerns have been raised about the massive amounts of private and public data used to train AI systems, that are often found leading to biased outcomes and reinforcing inequalities. Furthermore, internationally, organizations such as Amnesty International, have documented the dangers of AI tools with their use as a means of mass surveillance, and discrimination, among others.

Such discussions and attempts to identify and mitigate the risk of AI deployment have led to significant policy developments, such as the 'Artificial Intelligence Act' in the EU that establishes a common regulatory and legal framework for AI within the European Union, the 'Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence' presidential action by the US, as well as approaches such as the launch of the 'AI Safety Institute' in the UK.

Studies and discussions concerning the human rights policy and performance connected to AI is important and much needed to protect the different groups and communities in society. This blog attempts to analyze the human rights-related articles on AI published in an online database called Business and Human Rights Resource Center. Through text analysis in R, I will be exploring the following research questions in this blog:

1. What is the sentiment on AI and human rights across articles? Is it more positive or negative? How has the sentiment surrounding human rights and AI changed in the articles over time?
2. What are the main themes expressed in AI-related articles on the BHHRC website? Can we identify some distinct topics relating to AI and human rights?
3. How have the most relevant words relating to AI changed in the articles over time?

These research questions will help us understand the trend in the development of AI and human rights issues during the time period between 2018-2023.

Getting Text Data

The text data for this text analysis is sourced from the Business and Human Rights Resource Center(BHRRC) website. BHRRC is a global business and human rights knowledge hub/database that

consists of data on the human rights policy and performance of over 10,000 companies in over 180 countries. The website is designed to enable communities and NGOs get companies to address human rights concerns, and provide companies an opportunity to present their response to the concerns. For this analysis, I am interested in articles surrounding the Artificial Intelligence topic. Therefore, I filtered the BHRRC website to get AI-related articles (Source).

For the first step, I am creating an object to read in the CSV file consisting of the list of URL links for AI-related articles found in the BHRRC website. The time period covered by this text analysis is January of 2018 to May of 2024. This time period was chosen because of the web scrapping limitations of the website.

R Library

```r
```{r }
library(tidyverse)
library(pdftools)
library(tidytext)
library(rvest)
library(httr)
require(httr)
library(ldatuning)
library(tm)
library(topicmodels)
```
```

```r
```{r }
articles_url<-read_csv("Articles.csv") %>%
  pull(url)
```
```

For the next step, I created an empty list of dates of the article and used a for loop to extract the dates from the article URLs.

```{r }
dates<-list()


for(i in 1:length(articles_url)){
  dates[[i]] <-  articles_url[i] %>%
    read_html() %>%
    html_elements(".metadata-page p") %>%
    html_text()
}


dates


```

Then, I extracted the text data from each of the articles' URLs using an empty list and a for loop to populate the list. I added the article number and the date to the text extracted from the articles.

The for loop is used twice since the html_element to extract the text from the URL changed after a certain date.

```{r}
AI_list <- list()


for(i in 1:224){
  AI_list[[i]] <-  articles_url[i] %>%
    read_html() %>%
    html_elements(".richtext-block p") %>%
    html_text() %>%
    tibble() %>%
    rename(text = ".") %>%
    mutate(article_no = i) %>%
```

```
    mutate(date=dates[[i]])
}


for(i in 225:361){
  AI_list[[i]] <-  articles_url[i] %>%
    read_html() %>%
    html_elements(".html-block p") %>%
    html_text() %>%
    tibble() %>%
    rename(text = ".") %>%
    mutate(article_no = i) %>%
    mutate(date=dates[[i]])
}


AI_list
```

Data Preparation

We finally extracted the data for analysis. Once the data is extracted, the next step is to Data Preparation. In this step, will tidy the article to get the text data. For this step, we will first bind the rows of the tibbles to create one large tibble and then unnest it to create tokens from the text data.

```{r}
AI_tibble <- AI_list %>%
  bind_rows() %>%
  filter(text != "") %>%
  filter(text !="...")


tidy_AI_article <- AI_tibble %>%
```

```
  unnest_tokens(input = text,
                output = word) %>%
  anti_join(get_stopwords()) %>%
  separate(col=date, into=c('day', 'month','year'), sep = ' ')


```

Data Description

Now that we have tidied the data, we see that we now have 5 columns in our dataset. The first column is the article number, which indicates which article the word is from. The second to fourth columns are the day, month and year columns. The dataset initially had one column for the date and in the previous step, I separated the date column into three columns by extracting the day, month and year values. The final column for this dataset is the word column which consists of the tokens created in the previous step.

Exploratory Data Analysis

Let's do some data exploration with our dataset. I will answer a few questions with the EDA.

First, let's look at the years represented in our dataset. We want to confirm that the articles from 2018 to 2024 are included in our analysis.

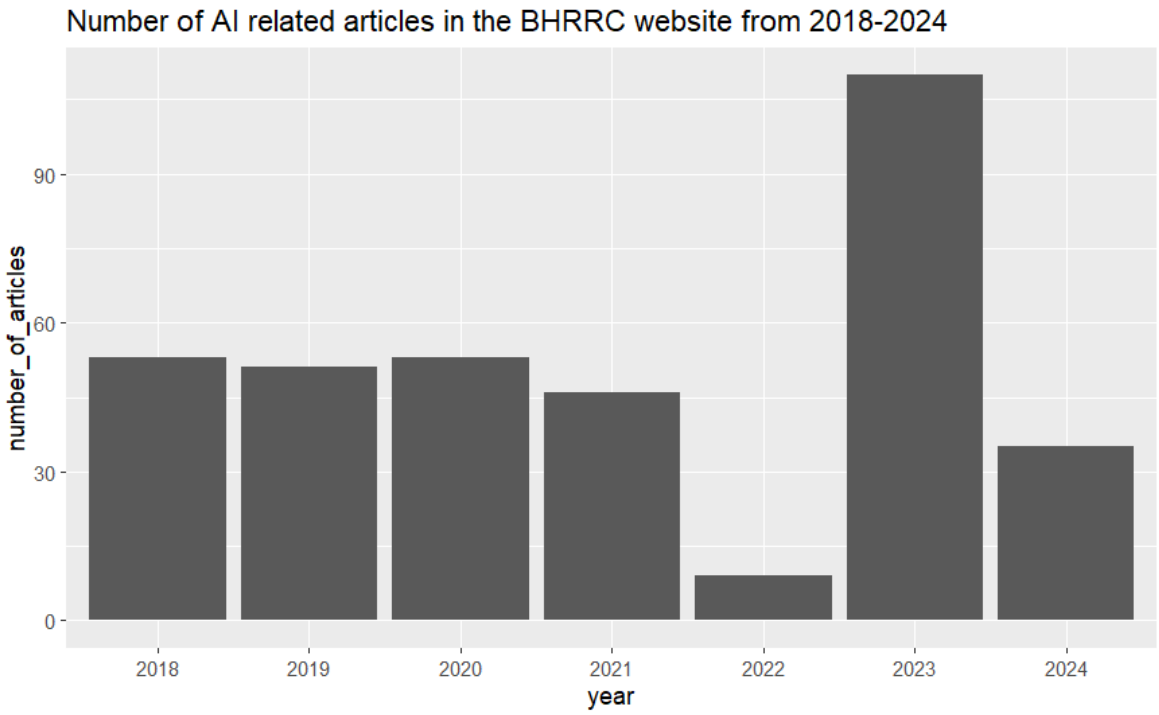
```{r}
tidy_AI_article$year %>%
  unique()
```

[1] "2024" "2023" "2022" "2021" "2020" "2019" "2018"

Now, let's look at the total number of articles published on the website per year. We want to see if there is an increasing or decreasing trend in discussions about Artificial Intelligence.

```{r}
tidy_AI_article%>%
  group_by(year) %>%
```

```
  summarize(number_of_articles=n_distinct(article_no)) %>%
  ggplot()+
  geom_col(aes(x=year, y=number_of_articles)) +
  ggtitle("Number of AI related articles in the BHRRC website from 2018-2024")
```

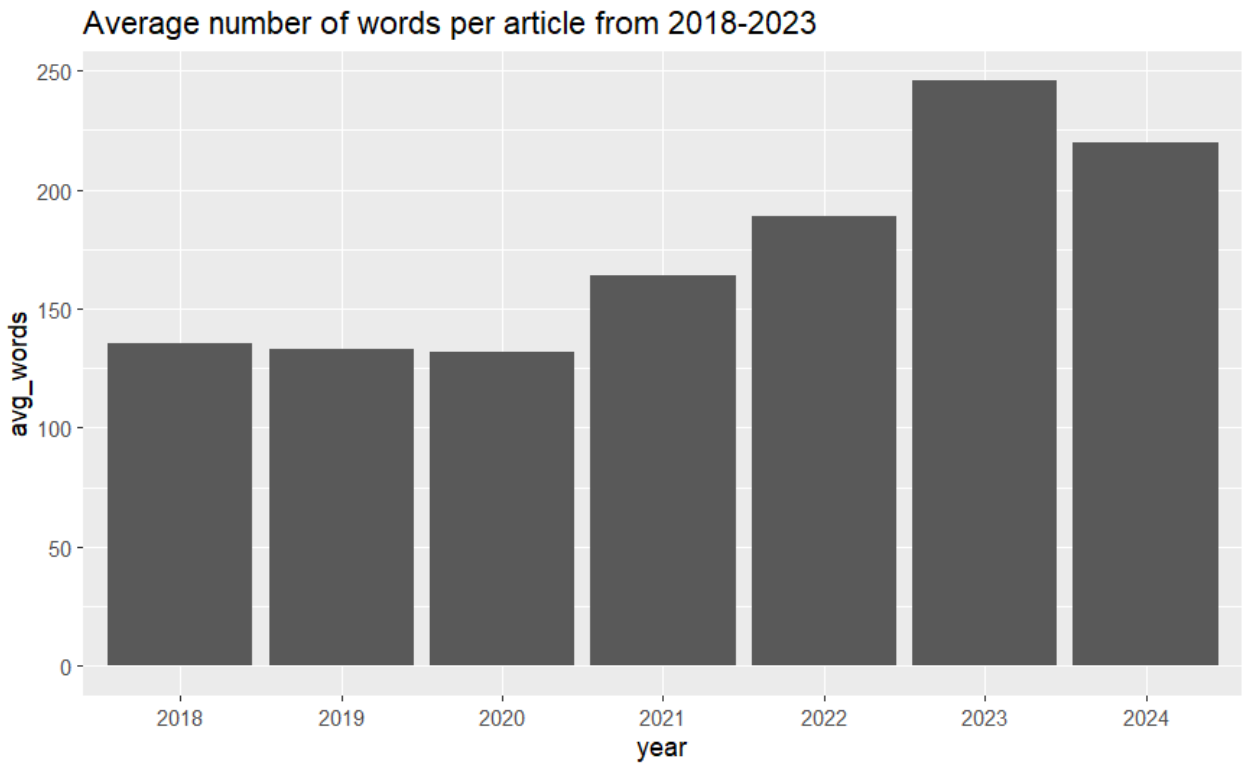


We see a pretty consistent trend for the number of articles published on the BHRRC website from 2018 to 2020. There is a slight dip in the total number of articles in 2021. An unusual observation is seen for the year 2022. For some reason, BHRRC website only has 9 AI-related articles for 2022. This might be because of a variety of reasons, one of which might be incorrect labeling of articles such as excluding the Artificial Intelligence topic although articles might have talked about it. I could not find any explanation on the BHRRC website for this situation.

We also observe a significantly higher number of articles about AI in 2023 compared to the rest of the year. This makes sense given the Generative AI boom and mass adoption by many big companies. The mass adoption of GenAI was also followed by a lot of discussion surrounding its ethical implications, which might explain the significant increase in the number of articles. 2024 has fewer number of articles since we only have articles from January 1st to May 15th for the year 2024.

Since there are not a lot of variables in our dataset, let's do one last data exploration and look at the yearly average number of words in articles.

```{r}
tidy_AI_article%>%
  group_by(year,article_no) %>%
  summarize(total_words=n()) %>%
  summarise(avg_words=mean(total_words)) %>%
  ggplot()+
  geom_col(aes(x=year, y=avg_words))+
  ggtitle("Average number of words per article from 2018-2023")
```

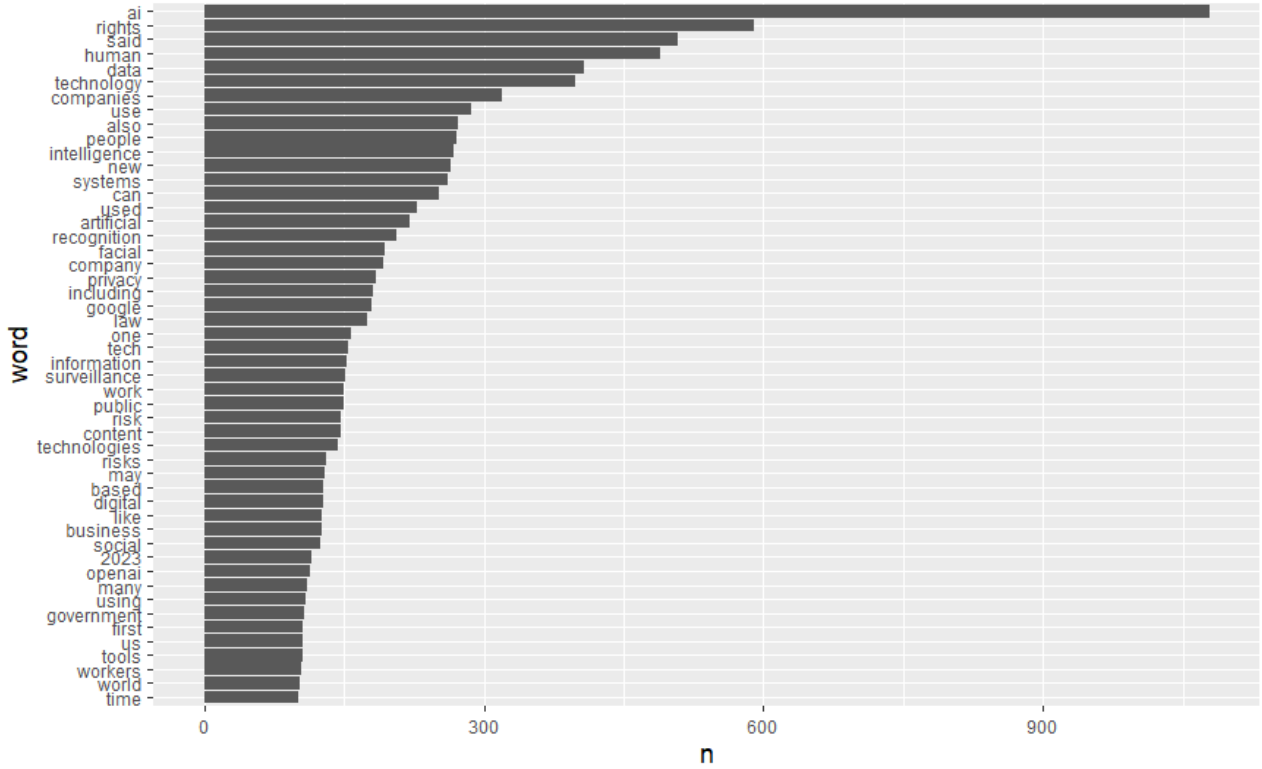


From this graph, we observe that the average words in the articles were similar from 2018 to 2020 but then it has an increasing trend over the years. Since BHRRC articles often provide a summary of external articles (original article sources are usually linked at the top of the BHRRC articles), it might be the case that they started providing a more detailed summary from 2021 onwards.

For the last step of data preparation before we proceed with the Text Analysis Modeling, let's look at the top words in the article.

```{r}
```

```
 tidy_AI_article%>%
  count(word,sort=T)%>%
  filter(n>100)%>%
  mutate(word=reorder(word,n))%>%
  ggplot()+
  geom_col(aes(x=n, y=word))+
  theme(axis.text=element_text(size=7))
```



Looking at the graph, there are some words that might not be useful for text analysis such as may, use, also, said, and so on. We might also want to look at tokenizing as bigrams and see if it generates better output.

Let's first tidy the original text data stored in an AI-tibble object and tokenize it as bigrams instead of words.

```{r}
```

```
AI_bigrams<-AI_tibble %>%
  unnest_tokens(input = text,
                output = bigram,
                token = "ngrams",
                n=2) %>%
  count(bigram) %>%
  arrange(desc(n)) %>%
  filter(!is.na(bigram)) %>%
  separate(col = bigram,
           into = c("w1","w2"),
           sep = " ") %>%
  anti_join(stop_words,
            by= c("w1" = "word")) %>%
  anti_join(stop_words,
            by = c("w2"="word"))
```

A tibble: 103,449 × 3

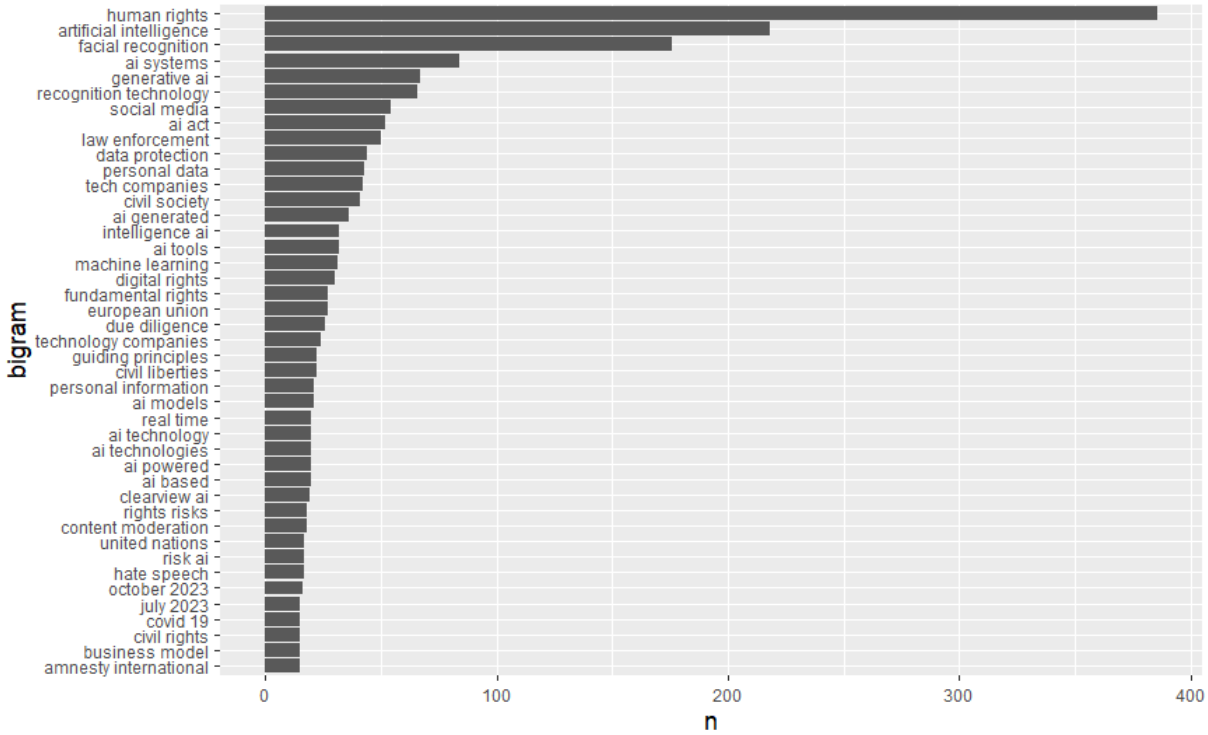
article_no <int>	date <chr>	bigram <chr>
1	29 Apr 2024	india global
1	29 Apr 2024	global new
1	29 Apr 2024	new technologies
1	29 Apr 2024	technologies in
1	29 Apr 2024	in automated
1	29 Apr 2024	automated social
1	29 Apr 2024	social protection
1	29 Apr 2024	protection systems
1	29 Apr 2024	systems can
1	29 Apr 2024	can threaten

1-10 of 103,449 rows Previous 1 2 3 4 5 6 ... 100 Next

Now that we have the bigrams, lets look at the most frequently occuring bigrams.

```
```{r}
```
```

```
AI_bigrams %>%
  mutate(bigram=reorder(paste(w1,w2),n)) %>%
  filter(n>14) %>%
  ggplot()+
  geom_col(aes(x=n, y=bigram))+
  theme(axis.text=element_text(size=7))
```



Based on the graphs, bigrams seem more useful as a unit of analysis for AI-related articles compared to single words. Therefore, we will proceed with bigrams for Text-Mining Analysis. Before that, there are some stop words I would like to remove from here such as dates so I will create a custom stop word list to do so.

```{r}
custom_stopwords<-tibble(word=c(
'2018','2019','2020','2021','2022','2023','2024','january',
'february','march','april','may','june','july','august','september','october'
,'november','december'), lexicon='ai')
```

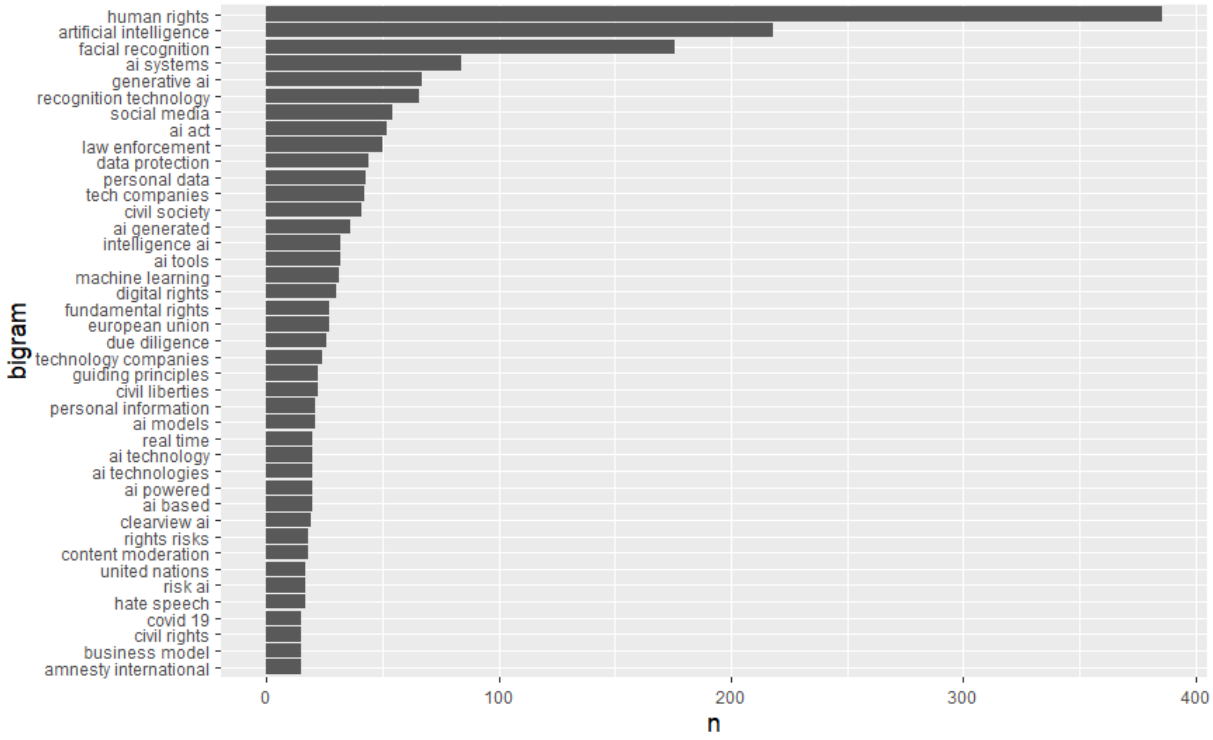
```
```

Let's remove these custom stopwords from our bigrams list.

```{r}
AI_bigrams<-AI_bigrams %>%
  anti_join(custom_stopwords,
            by= c("w1" = "word")) %>%
  anti_join(custom_stopwords,
            by = c("w2"="word"))
```

Let's take one last look at the count of most frequently-occurring words to make sure the bigrams are coherent and can be used for our analysis:

```{r}
AI_bigrams %>%
  mutate(bigram=reorder(paste(w1,w2),n)) %>%
  filter(n>14) %>%
  ggplot()+
  geom_col(aes(x=n, y=bigram))+
  theme(axis.text=element_text(size=7))
```



The bigrams seem coherent so we will move to the analysis step now.

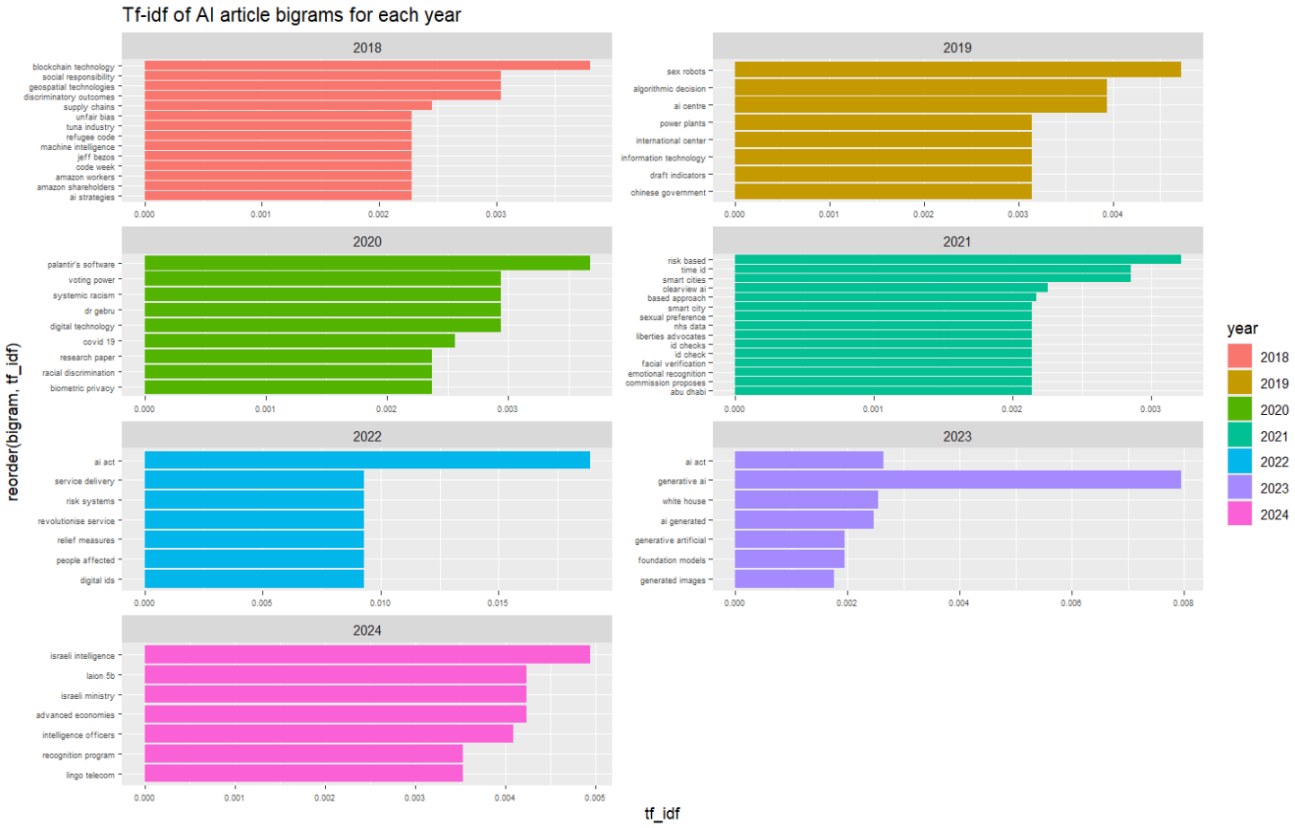
Text-Mining Analysis

TF-IDF Text Analysis

For the text-mining analysis, let us first look at the TF-IDF score to see the most relevant bigrams surrounding the AI and human rights articles for various years.

Tf-idf adjusts the frequency of a term based on how often it is used, thus, enabling us to look at the word importance. I have also faceted the figures by years for this statistic.

```
```{r}
AI_tibble %>%
  unnest_tokens(input = text,
                output = bigram,
                token = "ngrams",
                n=2) %>%
  separate(col=date, into=c('day', 'month','year'), sep = ' ') %>%
```
```

```
separate(col = bigram,
         into = c("w1","w2"),
         sep = " ") %>%
anti_join(stop_words,
          by= c("w1" = "word")) %>%
anti_join(stop_words,
          by = c("w2"="word")) %>%
anti_join(custom_stopwords,
          by= c("w1" = "word")) %>%
anti_join(custom_stopwords,
          by = c("w2"="word")) %>%
mutate(bigram=paste(w1,w2)) %>%
count(year,bigram) %>%
bind_tf_idf(term = bigram,
            document = year,
            n=n) %>%
group_by(year) %>%
top_n(7,
      wt=tf_idf) %>%
ggplot()+
geom_col(aes(x=tf_idf,
             y=reorder(bigram,tf_idf),
             fill=year))+
facet_wrap(~year,
           ncol=2,
           scales="free")+
theme( axis.text = element_text(size = 5.5))+
ggtitle(label="Tf-idf of AI article bigrams for each year")
```

```



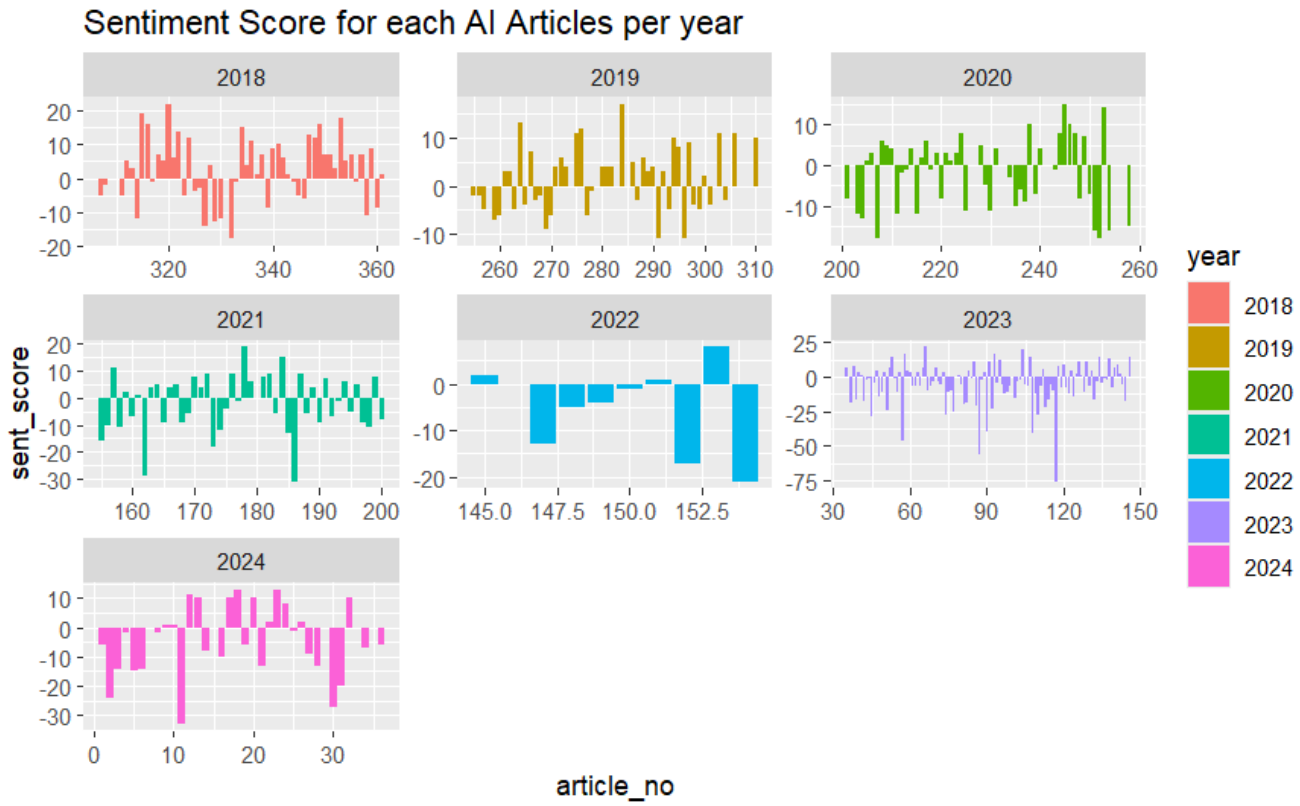

The column chart shows the top words in terms of relevance for the time period included in this analysis.

Some of the top words for 2018 are related to social responsibility, unfair bias, discriminatory outcomes, blockchain technology, and Amazon (Jeff Bezos, Amazon workers, and shareholders). When looking at the titles of articles from 2018, there seem to be various discussions about the potential benefits as well as the human rights risks, which might be why those words are higher in relevance. There were also many articles about Amazon's facial recognition technology and the opinions of the public and shareholders surrounding transparency.

For 2019, the bigrams that are most relevant seem to be sex robots and algorithmic decisions and we also see the Chinese government as one of the most relevant bigrams. There were articles discussing AI sex robots and how their law-making process should be shaped by human rights and ethical concerns. The Chinese government was one of the most relevant terms because of articles talking about facial scans being used to register for services.

For 2020, the most relevant discussions seem to be about racial discrimination, COVID-19, and biometric privacy because many articles around this time period were raising concerns over AI reinforcing racial bias and biometric/facial surveillance uses. For 2021 as well, there were many articles discussing facial recognition technology. NHS data was relevant since there were articles
```

about NHS's data deal with Palantir, which is a technology company that specializes in software platforms for big data analytics.

In 2022, most of the articles were about the EU's approach to the AI Act. In 2023, we see Generative AI-related terms as some of the most relevant terms, which makes sense given the boom in GenAI last year finally, 2024 had articles discussing Israel's use of AI in warfare, which is why it is among the most relevant words for this year.

We also notice some bigrams that do not immediately make sense such as sexual preference, code week, etc. This might need further exploration.

Sentiment Analysis

For the text-mining analysis, we also want to look at the sentiment surrounding Artificial Intelligence in these articles. We want to also see how sentiments have changed over the years.

Let's first create an object for the tokenized bigrams.

```
```{r}
AIData_bigrams<-AI_tibble %>%
  unnest_tokens(input = text,
                output = bigram,
                token = "ngrams",
                n=2) %>%
  separate(col=date, into=c('day', 'month','year'), sep = ' ')
```
```

Then, we can look at how sentiments surrounding AI are distributed across articles. I will use the afinn lexicon to get the sentiment scores. After getting the sentiment score, I will plot the sentiment scores across articles in a bar graph, faceting by years.

```
```{r}
AIData_bigrams %>%
  count(year,article_no,bigram) %>%
  arrange(desc(n)) %>%
  filter(!is.na(bigram)) %>%
  separate(col = bigram,
```
```

```
          into = c("w1","w2"),
          sep = " ") %>%
  inner_join(get_sentiments("afinn"),
            by = c("w2"="word")) %>%
  mutate(sent=if_else(w1 =="not",
                      value*-1,
                      value)) %>%
  group_by(year,article_no) %>%
  summarise(sent_score=sum(sent)) %>%
  ggplot()+
  geom_col(aes(x=article_no,
               y=sent_score,
               fill=year))+
  facet_wrap(~year,
             scales="free")+
  ggtitle("Sentiment Score for each AI Article per year")
```

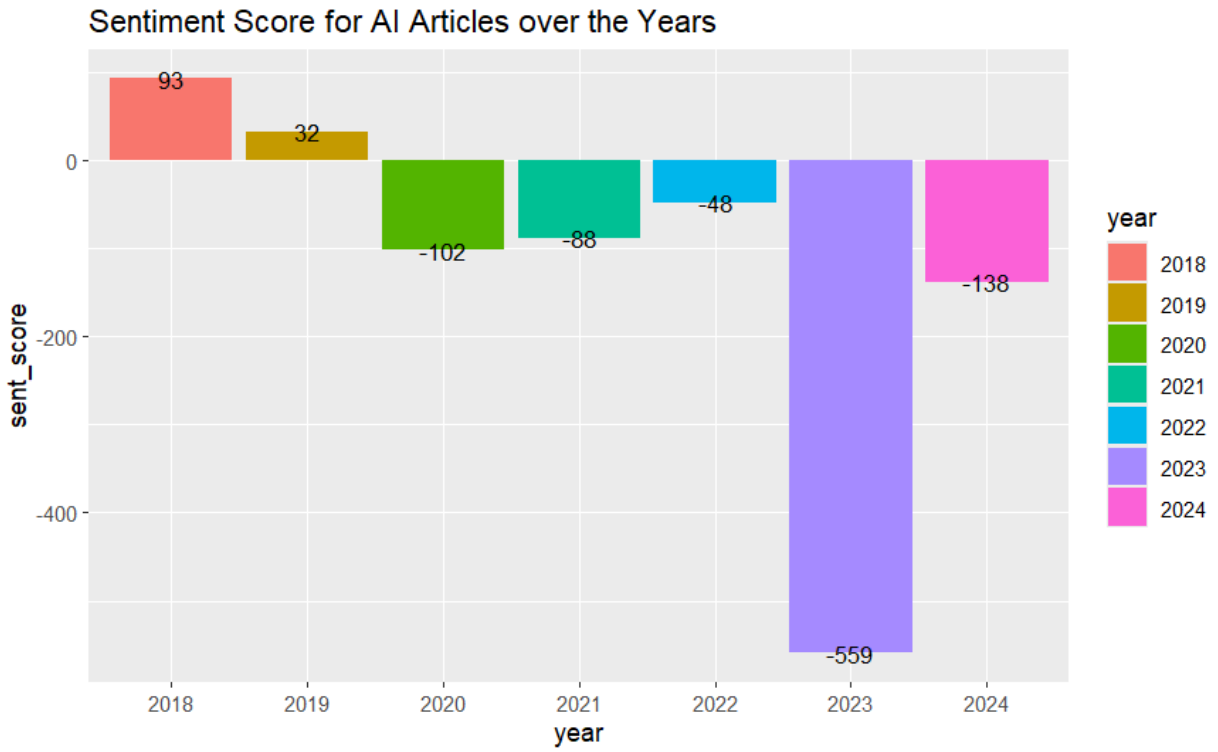


We observe that in the earlier years, the sentiment for the articles was more positive than negative. However, as time goes by, starting from 2020, more articles have an overall negative sentiment score than a positive one. In terms of the magnitude of the sentiment score value as well, we observe that for the positive articles, the maximum sentiment score seems to be around 20-30 but negative sentiment scores for articles go as low as -75 for 2023. We can say that discussions surrounding human rights policy and performance seem to have a more negative sentiment as the year increases.

Let's summarize the overall sentiment score by year and see how sentiments surrounding AI and human rights policy and performance changed over the years. I will do this by summing up the sentiment scores by year and plotting it on a bar graph.

```{r}
AIData_bigrams %>%
  count(year,bigram) %>%
  arrange(desc(n)) %>%
  filter(!is.na(bigram)) %>%
  separate(col = bigram,
           into = c("w1","w2"),
```

```
          sep = " ") %>%
  inner_join(get_sentiments("afinn"),
             by = c("w2"="word")) %>%
  mutate(sent=if_else(w1 =="not",
                      value*-1,
                      value)) %>%
  group_by(year) %>%
  summarise(sent_score=sum(sent)) %>%
  ggplot()+
  geom_col(aes(x=year,
               y=sent_score,
               fill=year))+
     geom_text(aes(x=year,
               y=sent_score,
               label = sent_score), size = 3.5)+
  ggtitle("Sentiment Score for AI Articles over the Years")
```

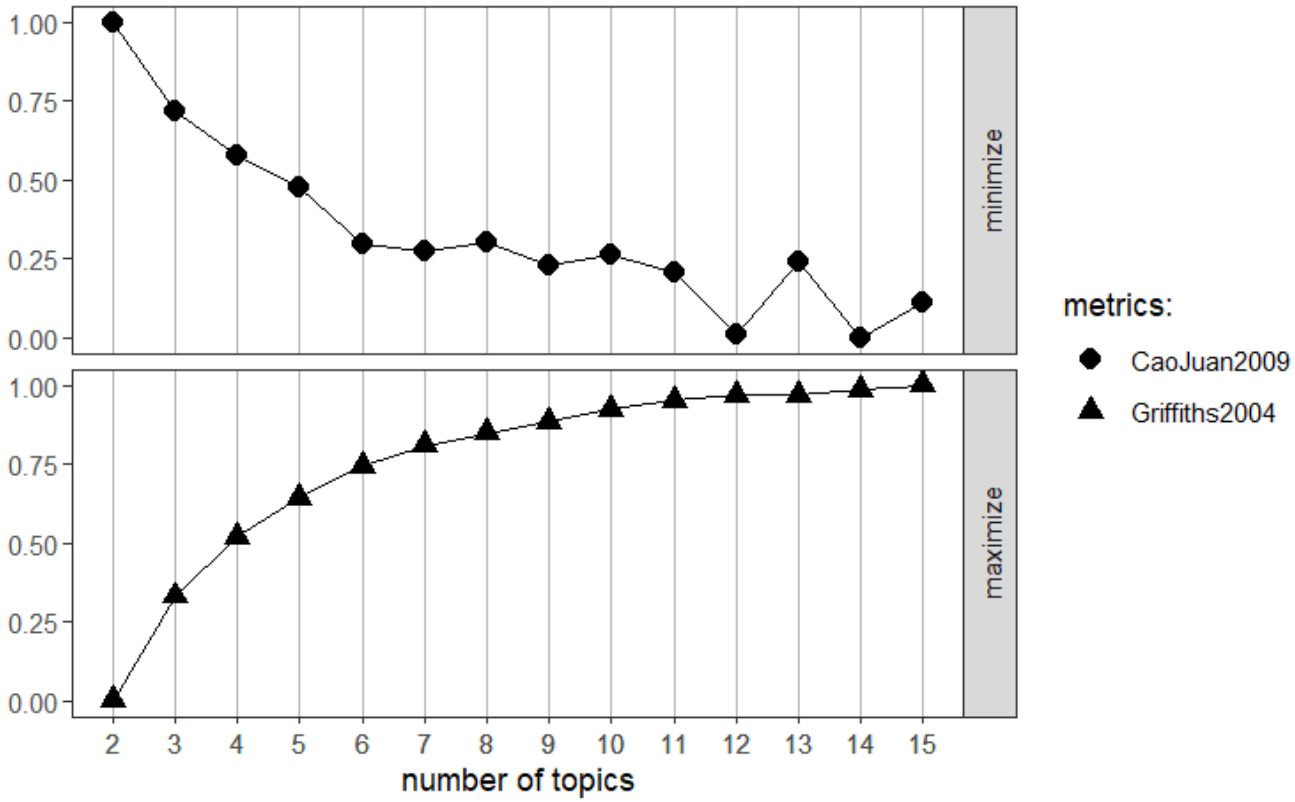


We can confirm the conclusions from the previous graph looking at the overall sentiment surrounding AI over the years. We see that as the year increases, the sentiment score goes from positive to negative. There does not seem to be a pattern but that might also be because of the lack of articles in the year 2022. The low sentiment score for 2022 can be attributed to a lesser number of articles in the website. Similarly, the BHRRC website had a significantly higher number of articles for 2023 compared to other years, which might be one of the reasons why the sentiment score for the year is so low. Looking at the pattern, the sentiment score for 2024 for the rest of the year is probably going to be negative as well.

Topic Modeling

For the final analysis, we will perform topic modeling to find out the main themes expressed in AI-related articles on the BHHRC website. To decide the number of topics, we will use the CaoJuan2009 and Griffiths2004 metrics. For CaoJuan2009, we choose the number of topics by minimizing the cosine distance between topics, while for Griffiths2004, we will look at the point where the graph flattens or stops increasing.

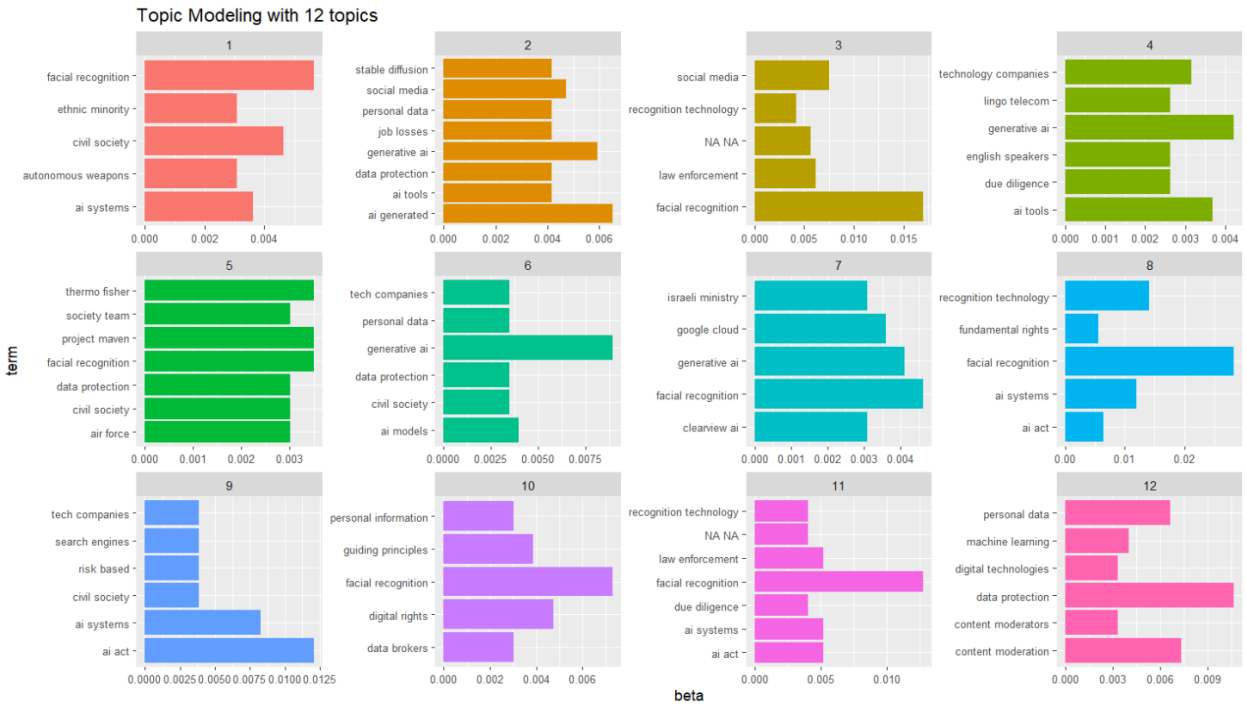
```
```{r}
AI_dtm <- AI_tibble %>%
  unnest_tokens(input = text,
```
```

```
                output = bigram,
                token = "ngrams",
                n=2) %>%
  separate(col=date, into=c('day', 'month','year'), sep = ' ') %>%
  count(article_no,bigram) %>%
  cast_dfm(document = article_no,
           term = bigram,
           value = n)


result <- ldatuning::FindTopicsNumber(
  AI_dtm,
  topics = seq(from = 1, to = 15, by = 1), ## running 1 to 15 topics to compare
and find the best number
  metrics = c("CaoJuan2009",  "Griffiths2004"),
  method = "Gibbs",
  control = list(seed = 77),
  verbose = TRUE
)
ldatuning::FindTopicsNumber_plot(result)
```



Based on both the CaoJun2009 and Griffiths2004 plots, we will choose 12 as the number of topics since we do not seem to be getting much additional information with more number of topics.

Let's now proceed with the topic modeling. First, we need to create a document term matrix with the tidy AI article data. Before that. I also created an additional stopwords list to prevent words like artificial intelligence and human rights from dominating the topics.

```
```{r}
additional_stopwords<-tibble(word=c(    'artificial    intelligence',    'human rights'), lexicon='ai')
```
```

```
```{r}
AIData_dtm <-  AI_tibble %>%
  unnest_tokens(input = text,
                output = bigram,
                token = "ngrams",
                n=2) %>%
```
```

```
  separate(col=date, into=c('day', 'month','year'), sep = ' ') %>%
  separate(col = bigram,
           into = c("w1","w2"),
           sep = " ") %>%
  anti_join(stop_words,
            by= c("w1" = "word")) %>%
  anti_join(stop_words,
            by = c("w2"="word")) %>%
  anti_join(custom_stopwords,
            by= c("w1" = "word")) %>%
  anti_join(custom_stopwords,
            by = c("w2"="word")) %>%
  mutate(bigram=paste(w1,w2)) %>%
 anti_join(additional_stopwords,
            by = c("bigram"="word")) %>%
  count(article_no,bigram) %>%
  cast_dtm(document = article_no,
           term = bigram,
           value = n)
AIData_dtm
```

<<DocumentTermMatrix (documents: 357, terms: 17589)>>
Non-/sparse entries: 21608/6257665
Sparsity : 100%
Maximal term length: 30
Weighting : term frequency (tf)

Now that we have the dtm, we will create an LDA topic model with 12 topics.

```
```{r}
AI_lda <- LDA(AIData_dtm,
```
```

```
              k = 12,
              control = list(seed = 1234))
AI_lda
```
A LDA_VEM topic model with 12 topics.

We will now look at the top bigrams for each of the 12 Topics using a faceted bar plot.

```{r}
AI_lda %>%
  tidy(matrix = "beta") %>%
  group_by(topic) %>%
  top_n(n = 10,wt=beta) %>%
  ggplot() +
  geom_col(aes(beta,term,fill = factor(topic)),
           show.legend = F) +
  facet_wrap(~factor(topic),scales= "free")+
  theme( axis.text = element_text(size = 6))+
  ggtitle(label="Topic Modeling with 12 topics")
```



Looking at the 12 topics, some topics seem to be distinct while the bigrams for some seem to overlap.

With this topic model, we are able to identify themes in some of the topics. For example, Topic 4 seems to be related to the 'Use of AI tools for language processing' since we see bigrams like lingo telecom and English speakers. Topic 6 seems to be related to 'Generative AI'. We often see concerns about personal data and data protection when talking about Generative AI being used by tech companies. Topic 10 might be related to articles that discuss guiding principles and laws for AI use since we see those words as well as words like digital rights and data brokers. Topic 11 seems to be related to recognition technology and law enforcement.

We also observe a lot of overlap in the words across topics, for example, words like law enforcement, AI systems, and facial recognition. It would be interesting to look at the topics by adding these to the additional stop words list, and observe whether topics become more distinct, however, these are relevant words surrounding the conversations about AI and human rights, so I am hesitant to remove them.

For extra analysis, let's also look at whether single words provide more distinct/identifiable topics compared to bigrams.

Topic Modeling using Word instead of Bigrams

```
```{r}

AIData_dtm2 <-  AI_tibble %>%
```
```

```
  unnest_tokens(input = text,
                output = word) %>%
  separate(col=date, into=c('day', 'month','year'), sep = ' ') %>%
  anti_join(get_stopwords()) %>%
  anti_join(custom_stopwords,
            by= c("word" = "word")) %>%
  count(article_no,word) %>%
  cast_dtm(document = article_no,
           term = word,
           value = n)


AIData_dtm2
```

<<DocumentTermMatrix (documents: 357, terms: 9879)>>
Non-/sparse entries: 47565/3479238
Sparsity : 99%
Maximal term length: 22
Weighting : term frequency (tf)

After creating the dtm, we will again create an LDA topic model with 12 topics.

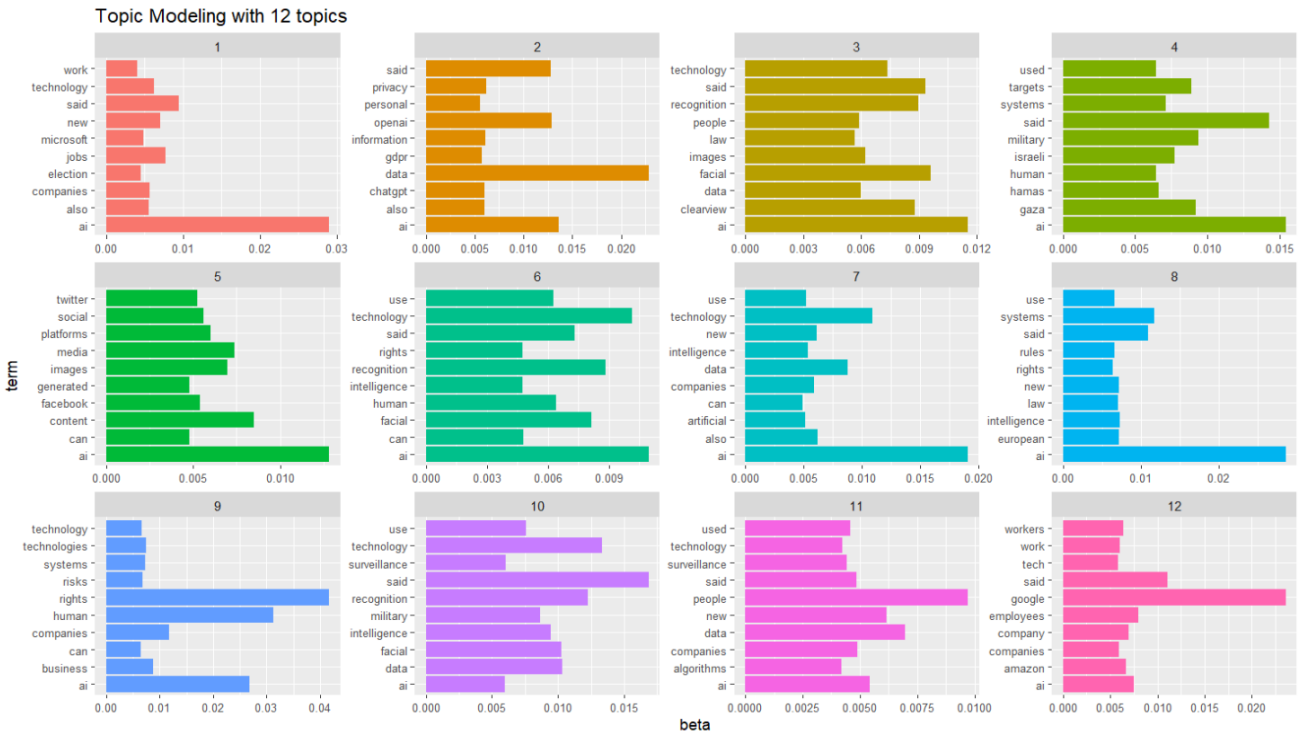
```{r}
AI_lda2 <- LDA(AIData_dtm2,
               k = 12,
               control = list(seed = 1234))
AI_lda2
```

A LDA_VEM topic model with 12 topics.

Let's look at the top words for each of the topics created using bar graph faceted by topics:

```{r}
AI_lda2 %>%
```

```
  tidy(matrix = "beta") %>%
  group_by(topic) %>%
  top_n(n = 7,wt=beta) %>%
  ggplot() +
  geom_col(aes(beta,term,fill = factor(topic)),
           show.legend = F) +
  facet_wrap(~factor(topic),scales= "free")+
  theme( axis.text = element_text(size = 8))+
  ggtitle(label="Topic Modeling with 12 topics")

```



Looking at the topics based on words, it seems that we indeed get more distinct and identifiable topics when using single words compared to bigrams. Topic 1 seems to be related to job creation and work due to AI and Topic 12 seems to be about workers and companies. Topic 2 is related to privacy, personal information, and laws such as the GDPR. Topic 4 is related to the use of AI tools in warfare. Topic 5 seems to be about AI and Social Media. Topic 9 is related to human rights and risks from AI technology. Topic 10 seems to be about different ways of using AI technology.

There are also overlapping words like before in this topic modeling as well. It might also be useful to remove some of the words by creating a stop words list (for example, verbs such as said, use, can, etc. that appear in some of the topics). It is important to note that the results might be different after removing the stop words.

Conclusion

We saw that the discussions surrounding AI and human rights have increased over the years, with many articles being written about it in 2023 specifically. This text analysis helped us discover how the sentiments surrounding AI and human rights issues have changed over time from 2018 to 2024. We observed a more negative sentiment as time progressed. We were also able to identify some topics/themes concerning the human rights issues on AI such as language processing, generative AI use, facial recognition and data protection, AI laws and guiding principles, and so on. This analysis can be used as a reference and stepping stone for a more granular text analysis for this topic.

***Note**: While extracting the dates and the text data, you might come across the Error: "Error in open. connection(x,"rb"): HTTP error 403.". This HTTP status code means that access to the requested resource is forbidden, and here it is mostly happening due to the website timing out as a result of network traffic. You might want to subset the URL list in the for loop during the web scrapping. It is also recommended to take some breaks in between web scrapping for each subset.*