

# Visualizing Social Debates on Inequality from a Large Social Media Knowledge Graph

Alkım Belevi, Evy Beijen, Mona Borth, Celeste Tacconi

2024-01-31

Social media platforms such as Twitter have become spaces for societal and political dialogue. However, existing strategies for analyzing such discourse often fail to capture the fine-grained opinions and contextual intricacies present in social media exchanges. We wished to address this limitation by employing the Observatory Knowledge Graph (OKG) (Blin, I., Stork, L., Spillner, L., Santagiustina, C.: OKG: A Knowledge Graph for Fine-grained Understanding of Social Media Discourse on Inequality. In: K-CAP '23: Knowledge Capture Conference 2023, Pensacola, FL, USA, pp. 1–13. ACM, New York (2023). DOI: <https://doi.org/10.1145/3587259.3627557>) as a base for our visualizations, capturing patterns and their intricacies related to inequality. The OKG enables fine-grained and contextual analyses as it captures semantic structures and is designed to aid researchers in understanding online discussions on inequalities along with relations between entities such as people, places, events, or organizations. To interact with this information, we used SPARQL queries to extract key characteristics, sentiments, and contextual information from the OKG. By visualizing the information, we aimed to extract meaningful insights from our dataset, bridging the gap between complex data and interpretable information.

The social media data about inequality was extracted from the Twitter platform through the “academicwitter” R library (Blin et al., 2023), using the Full-Archive API V2 with the query parameters “(inequality OR inequalities)lang:en.” This forms a sample of 62015 tweets published from May 30th to August 27th, 2020, and the OKG contains both metadata and linguistic information about the tweets of this sample. The visualizations of the OKG data are presented in this document, including brief interpretations and the R code used to create them.

## Setup

First, we save the `api_key`. Since we cannot make this key public, the code is hidden.

```
# Set API Key and headers
headers <- c(`Authorization` = paste0("Bearer ", api_key),
            `Accept` = "application/sparql-results+json")
# SPARQL Endpoint
sparql_endpoint <-
  'https://api.druid.datalegend.net/datasets/lisestork/OKG/services/OKG/sparql'
```

```
# Install and load necessary packages
```

```
#install.packages("httr")
#install.packages("jsonlite")
#install.packages("tidyr")
#install.packages("dplyr")
#install.packages("ggplot2")
```

```

#install.packages("wordcloud")
#install.packages("quanteda")
#install.packages("quanteda.textplots")
#install.packages("igraph")
#install.packages("stringr")
#install.packages("scales")
library(httr)
library(jsonlite)
library(tidyr)
library(dplyr)
library(ggplot2)
library(wordcloud)
library(quanteda)
library(quanteda.textplots)
library(igraph)
library(stringr)
library(scales)

```

```

# Set the SPARQL prefixes
query0 <- "
PREFIX obio: <https://www.w3id.org/okg/obio-ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX nee: <http://www.ics.forth.gr/isl/oe/core#>
PREFIX schema: <http://schema.org/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX earmark: <http://www.essepuntato.it/2008/12/earmark>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
PREFIX wsj: <https://w3id.org/framester/wsj/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
"

```

```

# Define a function that fetches all results to circumvent the max of 10000 rows

fetch_all_results <- function(endpoint, query, start, max_rows, step) {
  # for max_rows, put a very large number
  # start at 0
  # step works at 1000, code gets slower if it's smaller

  all_results <- list()
  offset <- start
  total_fetched <- 0

  repeat {
    remaining_rows = max_rows - total_fetched
    limit = min(step, remaining_rows)
    if (limit <= 0) break

    query_with_limit_offset <- paste(query, sprintf("LIMIT %d OFFSET %d", limit,
                                                    offset))
    response <- GET(endpoint, query = list(query = query_with_limit_offset),

```

```

        httr::add_headers(headers))

if (status_code(response) != 200) {
  stop("Error in fetching data:", " limit: ", limit, " offset: ", offset,
       content(response, "text"))
}

content_raw <- content(response, "text", encoding = "UTF-8")
data <- fromJSON(content_raw)
data_df <- as.data.frame(data$results$bindings, stringsAsFactors = FALSE)

if (nrow(data_df) == 0) { # No more results to fetch
  break
}

all_results[[length(all_results) + 1]] <- data_df
total_fetched <- total_fetched + nrow(data_df)
offset <- offset + step
}

return(bind_rows(all_results))
}

```

## 1. Word clouds of frequently mentioned entities

```

# Function to extract entity name from URI
get_entity_name <- function(uri) {
  # Extract the part after the last underscore and remove "ent_"
  gsub("ent_", "", sub(".*ent_(.*)", "\\1", uri))
}

# SPARQL Query
query <- "
SELECT ?entity WHERE {
  ?tweet schema:mentions ?entityMention .
  ?entity nif:anchorOf ?entityMention .
}
"

query <- paste(query0, query)

data_df3 <- fetch_all_results(sparql_endpoint, query, 0, 100000, 1000)

# Selecting relevant data
data_df3_adj <- data.frame(
  entity = data_df3$entity$value
)

# Getting the entity names in the "entity" column
data_df3_adj$entity <- sapply(data_df3_adj$entity, get_entity_name)

```

```
# Make wordcloud of the entities
entities <- count(data_df3_adj, entity)

wordcloud(words = entities$entity, freq = entities$n, col = "plum4",
          min.freq = 50, rot.per=0)
```

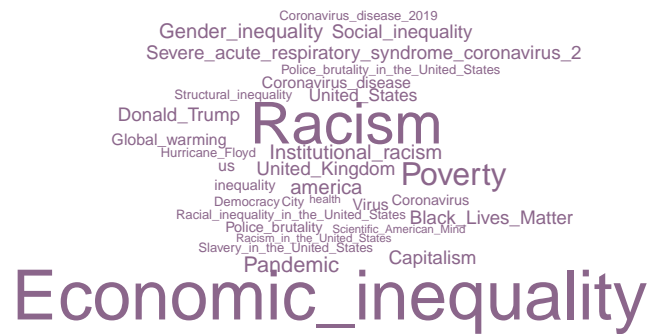


Figure 1: Wordcloud representing the most frequently mentioned entities in the dataset (>50 times).

The table below gives the counts for each entity shown in the wordcloud.

```
# Create a new dataframe with counts
entity_counts <- data_df3_adj %>%
  group_by(entity) %>%
  summarise(count = n())%>%
  arrange(desc(count))

# Show table with counts of entities that occur >50 times
knitr::kable(as.data.frame(entity_counts[1:31,]),
              caption = "Counts of entities mentioned >50 times")
```

Table 1: Counts of entities mentioned >50 times

entity	count
Economic_inequality	891
Racism	726
Poverty	335
Pandemic	186
Institutional_racism	180
Gender_inequality	168
america	164
Black_Lives_Matter	156
Capitalism	144
Donald_Trump	142
Severe_acute_respiratory_syndrome_coronavirus_2	141
Social_inequality	140
United_Kingdom	138
United_States	123
Global_warming	111
Virus	102
Coronavirus_disease	98
us	96
Police_brutality	95
Coronavirus	85
inequality	80
Coronavirus_disease_2019	73
Police_brutality_in_the_United_States	73
Hurricane_Floyd	67
City	62
Democracy	57
Racial_inequality_in_the_United_States	57
Slavery_in_the_United_States	56
Structural_inequality	56
Scientific_American_Mind	53
Racism_in_the_United_States	52

The automatic entity linking led to a lot of different entities being present in the dataset, while referring to the same or very similar things. An example of this are the entities connected to the COVID-19 like Coronavirus\_disease\_2019, Coronavirus\_disease, Severe\_acute\_respiratory\_syndrome\_coronavirus\_2, and Pandemic. To create a more accurate representation of how often this topic is mentioned in regard to others, we manually selected the entities (likely) related to COVID-19 that occur at least 20 times to be aggregated.



The entity COVID appears 714 times, the rest of the frequencies are as above.

*In Figure 1, COVID-19 related entities are scattered due to the automatic linking, which makes it hard to perceive the high frequency of COVID-19 mentions. In Figure 2, we aggregated the COVID-19 related entities to highlight the shortcomings of the linking process and show the high frequency of mentions related to COVID-19. Although we tend not to focus visually on COVID-19 related entities in Figure 1, it becomes obvious that the aggregated COVID entity is among the top 3 most frequently mentioned entities in Figure 2. This highlights the significance of visualization choices in shaping perception. It also illustrates how data processing can be leveraged for manipulation, particularly in emphasizing the magnitude of issues like COVID-19 (e.g., researchers could choose to aggregate entities if they want an issue to appear more relevant).*

## 2. Frequently mentioned entities and sentiment

```
# SPARQL Query
query <- "
SELECT ?entity
      (COUNT(?negative) AS ?countNegative)
      (COUNT(?neutral) AS ?countNeutral)
      (COUNT(?positive) AS ?countPositive)
      (COUNT(?t) AS ?totalCount)
WHERE {
  ?t schema:mentions ?entityMention ;
      obio:sentiment_label ?label .
  ?entity nif:anchorOf ?entityMention .

  OPTIONAL {
    ?t schema:mentions ?entityMentionN ;
        obio:sentiment_label ?labelN .
    ?entity nif:anchorOf ?entityMentionN .
    FILTER(?labelN = \"negative\")
    BIND(?t AS ?negative)
  }
  OPTIONAL {
    ?t schema:mentions ?entityMentionNu ;
        obio:sentiment_label ?labelNu .
    ?entity nif:anchorOf ?entityMentionNu .
    FILTER(?labelNu = \"neutral\")
    BIND(?t AS ?neutral)
  }
  OPTIONAL {
    ?t schema:mentions ?entityMentionP ;
        obio:sentiment_label ?labelP .
    ?entity nif:anchorOf ?entityMentionP .
    FILTER(?labelP = \"positive\")
    BIND(?t AS ?positive)
  }
}
GROUP BY ?entity
ORDER BY DESC(?totalCount)
LIMIT 10
"

query <- paste(query0, query)
```

```

# Make the GET request
response1 <- GET(sparql_endpoint, query = list(query = query),
                http::add_headers(headers))

# Check if the request was successful
if (status_code(response1) == 200) {
  content_raw1 <- content(response1, "text", encoding = "UTF-8")
  data1 <- fromJSON(content_raw1)
  data_df1 <- as.data.frame(data1$results$bindings)
} else {
  print(content(response1, "text"))
}

# Selecting relevant data
data_df1_adj <- data.frame(
  entity = data_df1$entity$value,
  totalCount = data_df1$totalCount$value,
  countNegative = data_df1$countNegative$value,
  countNeutral = data_df1$countNeutral$value,
  countPositive = data_df1$countPositive$value
)

# Getting the entity names in the "entity" column
data_df1_adj$entity <- sapply(data_df1_adj$entity, get_entity_name)

# Converting counts to numeric values
data_df1_adj$countNegative<-as.numeric(as.character(data_df1_adj$countNegative))
data_df1_adj$countNeutral<-as.numeric(as.character(data_df1_adj$countNeutral))
data_df1_adj$countPositive<-as.numeric(as.character(data_df1_adj$countPositive))
data_df1_adj$totalCount <- as.numeric(as.character(data_df1_adj$totalCount))

# Reshape the data using tidyr
data_df1_adj_long <- tidyr::gather(data_df1_adj, key = "sentiment",
                                value = "count", -entity, -totalCount)

# Add a new column to represent the sentiment of each count
data_df1_adj_long <- data_df1_adj_long %>%
  mutate(sentiment_total = ifelse(sentiment == "countNegative", "Negative",
                                ifelse(sentiment == "countNeutral", "Neutral", "Positive")))

# Order the data by totalCount in descending order
data_df1_adj_long <- data_df1_adj_long %>%
  arrange(desc(totalCount))

# Factorize the 'entity' column to enforce the desired order (descending)
data_df1_adj_long$entity <- factor(data_df1_adj_long$entity,
                                levels = unique(data_df1_adj_long$entity))

# Summarize the total counts for each entity
total_counts <- data_df1_adj_long %>%
  group_by(entity) %>%
  summarize(total_count = sum(count))

```



```
# Plotting
ggplot(data_df1_adj_long, aes(x = entity, y = count, fill = sentiment_total)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  geom_bar(data = total_counts, aes(x = entity, y = total_count, group = 1),
    stat = "identity", fill = "gray", alpha = 0.4,
    position = position_dodge(width = 0.8)) + # Overlay for total frequency
labs(title = "Most frequently mentioned entities including sentiments",
  x = "Entity",
  y = "Frequency", fill = "Sentiment") +
scale_fill_manual(values = c("Negative" = "#CC79A7", "Neutral" = "#999999",
  "Positive" = "#009E73")) +
scale_x_discrete(guide = guide_axis(n.dodge = 2)) + theme_bw() +
theme(axis.text=element_text(size=12), axis.title=element_text(size=14),
  title = element_text(size=16), legend.text = element_text(size=12))
```

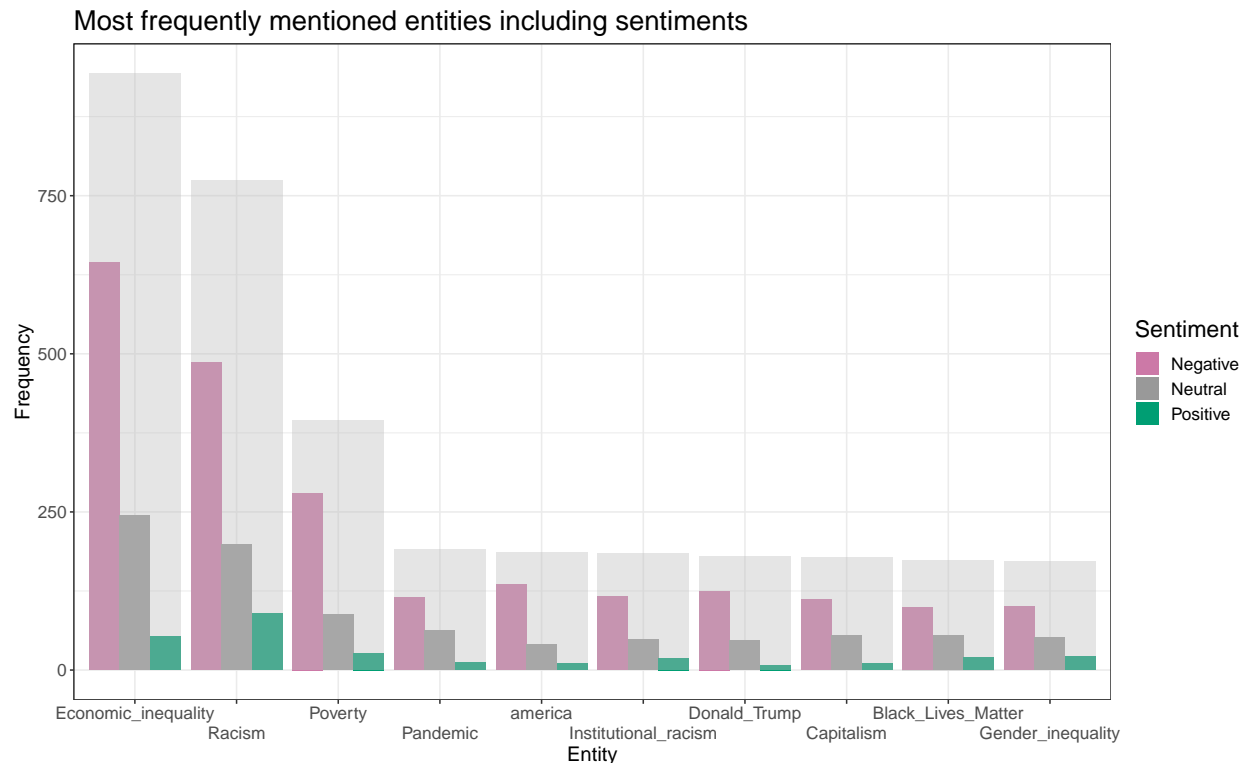


Figure 3: Bar chart of the top 10 most frequently occurring entities in the dataset, including distribution across the different sentiments.

Figure 3 clearly shows that negative tweets are the most prevalent among all entities in the top 10, followed by neutral tweets with positive tweets occurring least frequently. The time span of the dataset includes George Floyd being murdered by a police officer in Minneapolis (May 25th, 2020), which led to national and international Black Lives Matter (BLM) protests against racial injustice and police brutality. Thus, the high frequency of negative tweets, including entities such as “Racism,” “Institutional\_racism,” and “Black Lives Matter,” can potentially be partly linked to the BLM protests. Moreover, as the time span is during the COVID-19 pandemic, people were experiencing financial insecurity due to unemployment and high costs, which can be associated with the frequently occurring entities “Pandemic,” “Economic inequality,” and “Poverty.” Additionally, we can see that, out of these 10 entities, the entity “Racism” occurs most frequently in positive tweets. However, it should be noted that the accuracy of the sentiment labels is difficult to evaluate

since the actual texts of the tweets are not accessible.

### 3. Temporal analysis

#### Top 3 most frequently mentioned entities

```
# SPARQL Query
query <- "
SELECT ?entity ?day (COUNT(?post) AS ?nb_t)
WHERE {
  {
    SELECT ?post ?entity (xsd:date(?date) AS ?day)
    WHERE {
      ?post dc:created ?date .
      ?post schema:mentions ?entityMention .
      ?entity nif:anchorOf ?entityMention .
    }
    GROUP BY ?post ?entity ?day
    ORDER BY ?day
  }
}
GROUP BY ?day ?entity
"

query <- paste(query0, query)

data_df2 <- fetch_all_results(sparql_endpoint, query, 0, 100000, 1000)

# Selecting relevant data
data_df2_adj <- data.frame(
  entity = data_df2$entity$value,
  date = data_df2$day$value,
  count = data_df2$nb_t$value
)

# Getting the entity names in the "entity" column
data_df2_adj$entity <- sapply(data_df2_adj$entity, get_entity_name)

# Convert the "date" column to a proper date format
data_df2_adj$date <- as.POSIXct(data_df2_adj$date, format = "%Y-%m-%dZ",
                                tz = "UTC")

# Creating all combinations of entities and dates
all_entities <- unique(data_df2_adj$entity)
all_dates <- unique(data_df2_adj$date)
all_combinations <- expand.grid(
  entity = all_entities,
  date = all_dates
)

# Merging with the existing data frame to fill in missing combinations
data_df2_adj <- merge(all_combinations, data_df2_adj, by = c("entity", "date"),
```

```

all.x = TRUE)

# Filling NAs with 0 for count column
data_df2_adj$count[is.na(data_df2_adj$count)] <- 0

# Convert the "count" column to numeric
data_df2_adj$count <- as.numeric(data_df2_adj$count)

# Subset with the entities we want to show (top 3)
subset_df2 <- data_df2_adj %>% filter(entity %in% c("Economic_inequality",
"Racism", "Poverty"))

# Colors suitable for people with color blindness
cbPalette <- c("#D55E00", "#56B4E9", "#CC79A7", "#999999", "#009E73", "#F0E442",
"#0072B2", "#E69F00")

# Plot the counts over time
ggplot(subset_df2, aes(x = date, y = count, color = entity)) +
  geom_line() +
  labs(title = "Frequency of top 3 most frequently mentioned entities",
x = "Date (in 2020)",
y = "Frequency",
color = "Entity") +
  scale_colour_manual(values=cbPalette) + theme_bw() +
  theme(axis.text=element_text(size=12), axis.title=element_text(size=14),
title = element_text(size=16), legend.text = element_text(size=12))

```

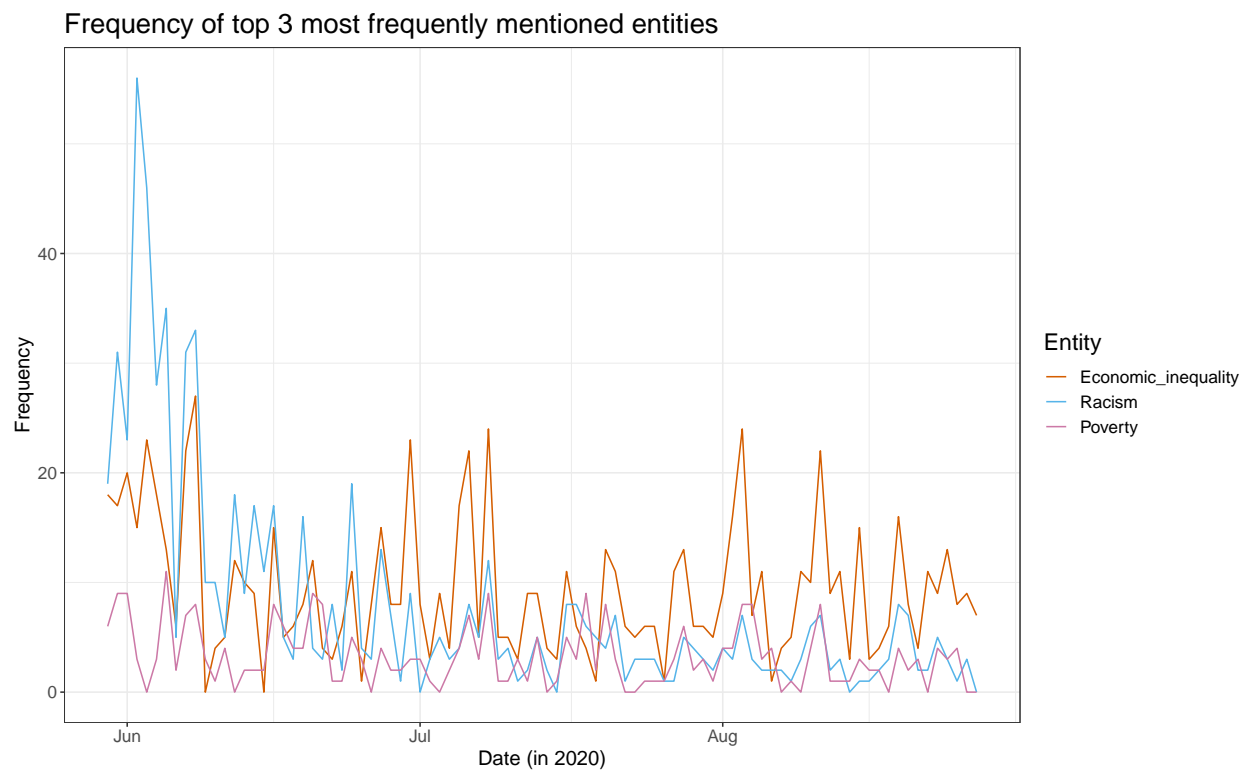


Figure 4: Line chart displaying the frequency of entity mentions over time for the top 3 most frequently occurring entities in the dataset.

*This line chart shows how often each of the top 3 entities was mentioned on each day in the time span of collection of the dataset. The entities “Economic\_inequality” and “Poverty” consistently stay within a range of 0 to 30 mentions per day throughout the entire period. In contrast, “Racism” shows a sharp increase in mentions at the beginning of June, followed by a gradual decline in daily mentions. The peak of the entity “Racism” can possibly be traced back to George Floyd’s murder and the events that followed. More specifically, on the 2nd of June, the collective protest “Blackout Tuesday” took place. Initially organized by the music industry, other businesses took part in producing blacked-out, silent, and minimal broadcasts in response to the murders of George Floyd, Breonna Taylor, and Ahmaud Arbery. Users participated in the protest on social media by sharing a black square photo with the hashtag #blackouttuesday.*

## Explaining the peak for the entity “Racism”

We manually selected all entities in the top 25 most frequently occurring entities that could be related to George Floyd’s murder and the response that followed. These were selected to substantiate our hypothesis that the peak in mentions of racism was related to George Floyd’s murder.

```
# Subset with the entities we want to show (related to George Floyd)
subset_df2 <- data_df2_adj %>% filter(entity %in% c("Institutional_racism",
  "Black_Lives_Matter", "Hurricane_Floyd", "Police_brutality",
  "Police_brutality_in_the_United_States"))

# Plot the counts over time
ggplot(subset_df2, aes(x = date, y = count, color = entity)) +
  geom_line() +
  labs(title = "Frequency of entity mentions (likely) related to
    George Floyd's murder in May 2020",
    x = "Date",
    y = "Frequency",
    color = "Entity") +
  scale_colour_manual(values=cbPalette) + theme_bw() +
  theme(axis.text=element_text(size=12), axis.title=element_text(size=14),
    title = element_text(size=16), legend.text = element_text(size=12))
```

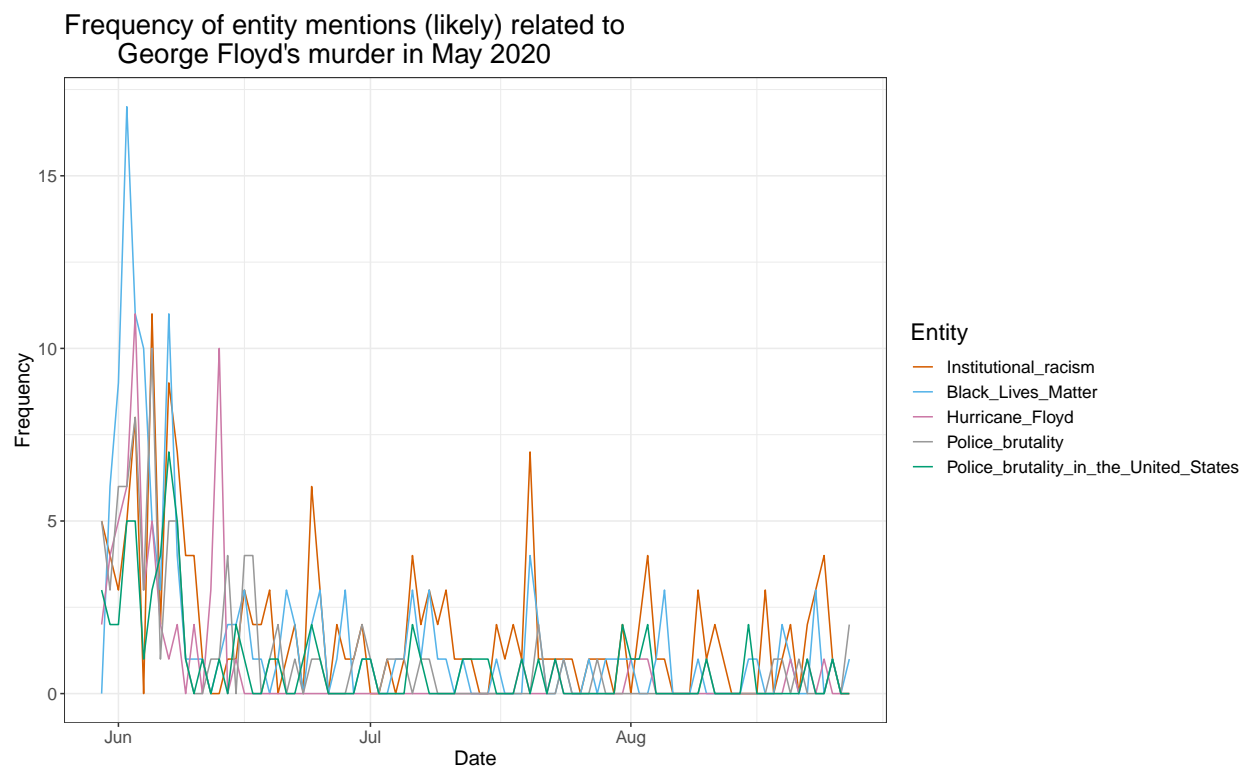


Figure 5: Line chart displaying the frequency of entity mentions that are (likely) related to George Floyd’s murder (by a police officer in Minneapolis on May 25th, 2020). N.B.: The entity “Hurricane\_Floyd” was likely wrongly linked in the automated entity extraction process, as Hurricane Floyd happened in 1999. It is plausible that tweets linked to the entity “Hurricane\_Floyd” are actually referring to George Floyd.

*We can observe that in the aftermath of George Floyd’s murder, the “Black\_Lives\_Matter” entity skyrocketed at the end of May/beginning of June in line with the upward trend in all the other entities. However, despite the initial frequent mention of entities related to racism, we observe a decrease in popularity in the entities after the initial peak.*

## Entities related to COVID-19

We used the same aggregation of COVID-related entities as in the word cloud (Figure 2), i.e. the entity “COVID” is made up of “Pandemic”, “Severe\_acute\_respiratory\_syndrome\_coronavirus\_2”, “Virus”, “Coronavirus\_disease”, “Coronavirus”, “Coronavirus\_disease\_2019”, “global\_pandemic\_response”, and “COVID-19\_pandemic”.

```
# Subset with the entities we want to show (related to COVID)
subset_df2 <- data_df2_adj %>% filter(entity %in% c("Pandemic",
  "Severe_acute_respiratory_syndrome_coronavirus_2", "Virus",
  "Coronavirus_disease", "Coronavirus", "Coronavirus_disease_2019",
  "global_pandemic_response", "COVID-19_pandemic"))

# Using aggregate to get total counts for each date
total_counts <- aggregate(count ~ date, data = subset_df2, sum)

# Plot the counts over time
ggplot(total_counts, aes(x = date, y = count)) +
  geom_line(color="plum4") +
  labs(title = "Frequency of entity mentions related to the topic of COVID-19",
    x = "Date", y = "Frequency") + theme_bw() +
  theme(axis.text=element_text(size=12), axis.title=element_text(size=14),
    title = element_text(size=16), legend.text = element_text(size=12))
```

Frequency of entity mentions related to the topic of COVID-19

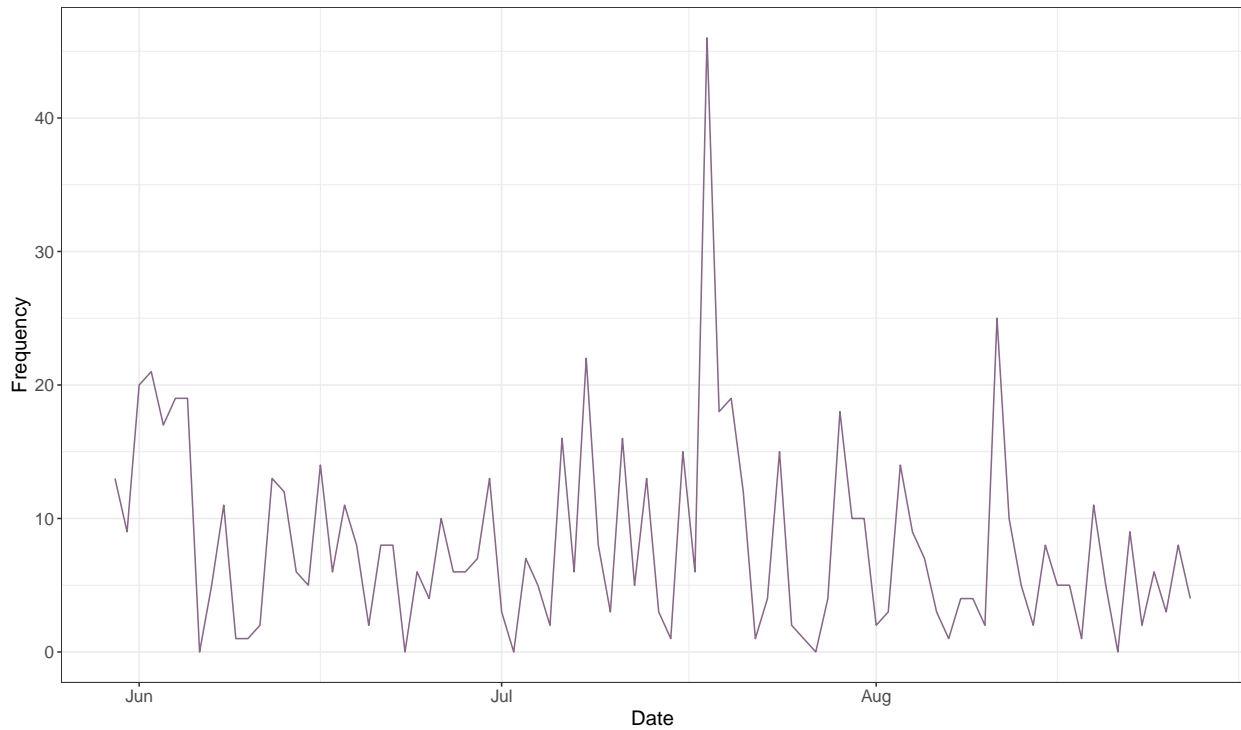


Figure 6: Line chart displaying the frequency of entity mentions related to COVID-19 over time.

*The abundant number of unique entries related to COVID-19 created difficulties in measuring the frequency of entities mentioned related to COVID-19. Thus, in this visualization, we manually aggregated COVID-19 related entities to show the temporal trend of entities referring to the COVID-19 pandemic. Throughout the time period, we can observe a steady trend with a peak on July 18th. Note that this day is Nelson Mandela Day, and mentions of #nelsonmandeladay peaked as well. However, our research has not revealed how this is connected to the peak of COVID-19 related entities on that day.*

## 4. Frequently co-occurring entities

```
# SPARQL Query
query <- "
SELECT ?ent_1 ?url_1 ?ent_2 ?url_2
(COUNT(?t) as ?nb_t) WHERE {
  ?t schema:mentions ?ent_m_1, ?ent_m_2 .
  ?ent_1 nif:anchorOf ?ent_m_1 .
  ?ent_2 nif:anchorOf ?ent_m_2 .
  OPTIONAL {?ent_1 nee:hasMatchedURL ?url_1 .}
  OPTIONAL {?ent_2 nee:hasMatchedURL ?url_2 .}
  FILTER (STR(?ent_1) < STR(?ent_2))
}
GROUP BY ?ent_1 ?url_1 ?ent_2 ?url_2
ORDER BY DESC(?nb_t)
"
query <- paste(query0, query)
```



```

# Make the GET request
response <- GET(sparql_endpoint, query = list(query = query),
               http::add_headers(headers))

# Check if the request was successful
if (status_code(response) == 200) {
  content_raw <- content(response, "text", encoding = "UTF-8")
  data <- fromJSON(content_raw)
  data_df4 <- as.data.frame(data$results$bindings)
} else {
  print(content(response, "text"))
}

data_df4$ent_1$value <- sapply(data_df4$ent_1$value, URLdecode)
data_df4$ent_2$value <- sapply(data_df4$ent_2$value, URLdecode)

data_df4$nb_t$value <- as.numeric(as.character(data_df4$nb_t$value))

# Filter rows where nb_t$value is greater than 8
data4 <- data_df4[data_df4$nb_t$value > 9, ]

# Get the entity names in the "entity" column
data4$ent_1 <- sapply(data4$ent_1, get_entity_name)
data4$ent_2 <- sapply(data4$ent_2, get_entity_name)

d <- data.frame(
  ent_1 = data4$ent_1,
  ent_2 = data4$ent_2,
  nb_t = data4$nb_t
)

d2 <- data.frame(
  ent_1 = d$ent_1.value,
  ent_2 = d$ent_2.value,
  nb_t = d$nb_t.value
)

query0 <- "
SELECT ?entity (COUNT(?t) as ?nb_t) WHERE {
  ?t schema:mentions ?entityMention .
  ?entity nif:anchorOf ?entityMention .
}
GROUP BY ?entity
ORDER BY DESC(?nb_t)
"

query <- paste(query0, query)

# Make the GET request
response <- GET(sparql_endpoint, query = list(query = query),
               http::add_headers(headers))

# Check if the request was successful

```

```

if (status_code(response) == 200) {
  content_raw <- content(response, "text", encoding = "UTF-8")
  data5 <- fromJSON(content_raw)
  data_df_entities <- as.data.frame(data5$results$bindings)
} else {
  print(content(response, "text"))
}

data_df_entities$entity$value <- sapply(data_df_entities$entity$value,
                                         URLdecode)

entities_weights <- data.frame(
  entity = data_df_entities$entity$value,
  nb_t = data_df_entities$nb_t$value
)

# Create a graph
g <- graph_from_data_frame(d2, directed = FALSE)

# Assign edge weight
E(g)$weight <- d2$nb_t

vertices <- V(g)$name
vertices <- sapply(vertices, URLdecode)

escape_special_chars <- function(string) {
  string <- gsub("[.\\\\\\+*?\\\\\\\\\\^\\\\\\$(){}=!<>|:-]", "\\\\\\\\\\\1", string,
                perl = TRUE)
  return(string)
}

weights <- list()
len_w = 1

for (vertex in vertices) {
  escaped_vertex <- escape_special_chars(vertex)
  pattern <- paste0("ent_", escaped_vertex, "$")
  weights[len_w] <- entities_weights[str_detect(entities_weights$entity,
                                                pattern), "nb_t"]

  len_w <- len_w + 1
}

weights <- as.numeric(weights)
weights <- log(weights)

# Assign the node sizes
V(g)$size <- weights

plot(g,
     edge.color = "grey",
     edge.width = E(g)$weight/12,
     vertex.size = V(g)$size,
     vertex.label.dist = 0.7,

```

```
knitr::include_graphics("./Images/Network_Zoom.png")
```

[illegible]

*In this network graph, the nodes represent entities, and the edges represent co-occurrences between the entities. The width of the edge is determined by the frequency of co-occurrence. The group of entities at the bottom of the visualization forms a cluster where we can identify the frequently mentioned entities like “Economic inequality,” “Racism,” and “Poverty.” This affirms the centrality of these entities in the dataset.*

We include the code to obtain the data, but include the visualizations as images since we edited them for aesthetic purposes.

19

```

ORDER BY DESC(?nb_t)
"

query <- paste(query0, query)

# Make the GET request
response7 <- GET(sparql_endpoint, query = list(query = query),
                http::add_headers(headers))

# Check if the request was successful
if (status_code(response7) == 200) {
  content_raw7 <- content(response7, "text", encoding = "UTF-8")
  data7 <- fromJSON(content_raw7)
  data_df7 <- as.data.frame(data7$results$bindings)
} else {
  print(content(response7, "text"))
}

```

```
knitr::include_graphics("../Images/Top10_retweets.png")
```

## Top 10 entities among all tweets vs. among tweets with >25 retweets

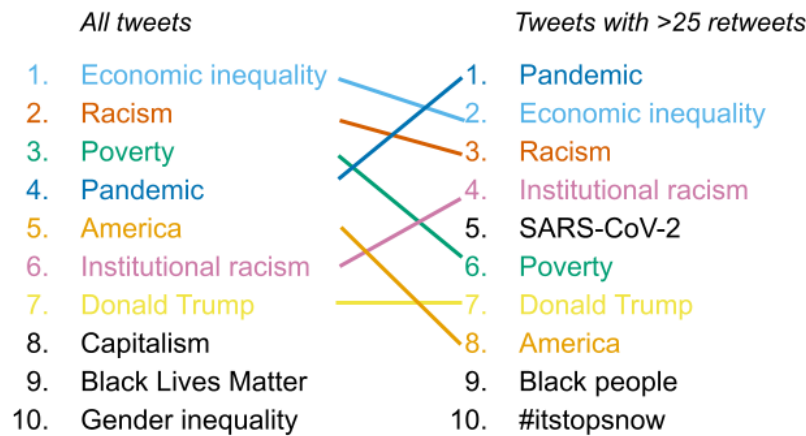


Figure 8: Top 10 most frequently occurring entities in the entire dataset, and in a subset of the dataset only containing tweets with >25 retweets. Entities that only appear in one of the groups are colored black.

*This color-coded visualization shows that “Economic inequality,” “Racism,” and “Pandemic” appear in the top 5 of both lists, suggesting that these issues are consistently significant topics in Twitter discourse about inequality. The prominence of topics related to the COVID-19 pandemic in the subset with >25 retweets indicates how high-impact events can dominate discussions with a larger reach. Moreover, in the same subset, Black people and the hashtag #itstopsnow, promoting a campaign fighting against the systemic problem of sexual harassment, gain additional visibility. N.B.: The decision to only consider the entities placed in the top*

10 positions could lead to the exclusion of potentially significant entities that fall just outside the designated range, such as those positioned at spot 11. Hence, one should be cautious in drawing strong conclusions from these visualizations.

```
# SPARQL Query
query <- "
SELECT ?entity (COUNT(?tweet) as ?nb_t) WHERE {
    ?tweet schema:mentions ?entityMention ;
    obio:nb_repost ?retweets ;
    obio:sentiment_label ?sentiment .
    ?entity nif:anchorOf ?entityMention .
FILTER (?sentiment = \"negative\")
}
GROUP BY ?entity
ORDER BY DESC(?nb_t)
\"# for neutral/positive tweets, change FILTER statement accordingly

query <- paste(query0, query)

# Make the GET request
response8 <- GET(sparql_endpoint, query = list(query = query),
                http::add_headers(headers))

# Check if the request was successful
if (status_code(response8) == 200) {
    content_raw8 <- content(response8, "text", encoding = "UTF-8")
    data8 <- fromJSON(content_raw8)
    data_df8 <- as.data.frame(data8$results$bindings)
} else {
    print(content(response8, "text"))
}

knitr::include_graphics("../Images/Top10_sentiment.png")
```

## Top 10 entities per sentiment label

<i>Negative tweets</i>	<i>Neutral tweets</i>	<i>Positive tweets</i>
1. Economic inequality	1. Economic inequality	1. Racism
2. Racism	2. Racism	2. Economic inequality
3. Poverty	3. Poverty	3. Poverty
4. America	4. Pandemic	4. Gender inequality
5. Institutional racism	5. Scientific American Mind	5. Black Lives Matter
6. Pandemic	6. Gender inequality	6. Institutional racism
7. Donald Trump	7. SARS-CoV-2	7. Coronavirus disease
8. Gender inequality	8. Black Lives Matter	8. Podcast
9. Capitalism	9. Institutional racism	9. United Kingdom
10. Social inequality	10. United Kingdom	10. Web conferencing

Figure 9: Top 10 most frequently occurring entities in subsets of the dataset containing only negative, neutral, or positive tweets, respectively. Entities that only appear in one of the groups are colored black.

The visualization shows that negative tweets about inequality were often tied to the challenges of 2020, including economic inequality, racism, poverty, and the impact of the pandemic. Discussions surrounding institutional racism, social inequality, and capitalism possibly reflect heightened tensions over social and systemic inequality. As for negative discourse, entities such as “Economic inequality”, “Racism”, “Poverty”, and “Pandemic” are subjects of neutral discourse. The entity “Scientific American Mind” refers to an American popular science magazine that seemingly posted a number of neutral tweets about inequality in the studied time period. The top 10 for positive tweets show that discussions around racism, economic inequality, and poverty also took on a positive tone. It is possible that such conversations were addressing solutions, progress, or collective resilience, for instance.

## 6. Statistics per tweet

```
# SPARQL Query
query <- "
SELECT ?following ?followers ?likes ?retweets ?label ?polarity WHERE {
    ?tweet sioc:has_creator ?creator;
    obio:nb_repost ?retweets ;
    obio:nb_like ?likes ;
    obio:sentiment_label ?label ;
    obio:polarity_score ?polarity .
    ?creator obio:follower ?followers ;
    obio:following ?following .
}
"
```

```

query <- paste(query0, query)

data_df4 <- fetch_all_results(sparql_endpoint, query, 0, 100000, 1000)

# Selecting relevant data
data_df4_adj <- data.frame(
  followers = data_df4$followers$value,
  following = data_df4$following$value,
  retweets = data_df4$retweets$value,
  likes = data_df4$likes$value,
  sentiment = data_df4$label$value,
  polarity = data_df4$polarity$value)

# Converting counts to numeric values
data_df4_adj$followers <- as.numeric(as.character(data_df4_adj$followers))
data_df4_adj$following <- as.numeric(as.character(data_df4_adj$following))
data_df4_adj$retweets <- as.numeric(as.character(data_df4_adj$retweets))
data_df4_adj$likes <- as.numeric(as.character(data_df4_adj$likes))
data_df4_adj$polarity <- as.numeric(as.character(data_df4_adj$polarity))

# Get random subset of 5% of the tweets
rand_df2 <- data_df4_adj[sample(nrow(data_df4_adj), size=3926), ]

ggplot(rand_df2) +
  geom_point(aes(x = 1+followers, y = 1+following, size = likes,
                 color=factor(sentiment))) +
  scale_x_log10(labels = label_number()) +
  scale_y_log10(labels = label_number()) +
  ggtitle(" Likes and sentiment per tweet by author's number of followers and following") +
  scale_color_manual(values = c("positive" = "#009E73", "negative" = "#CC79A7",
                                "neutral" = "grey")) +

  labs(x = "Followers",
       y = "Following",
       color = "Sentiment",
       size = "Number of likes") +
  geom_abline(intercept=0, slope=1) + theme_bw() +
  theme(axis.text=element_text(size=12), axis.title=element_text(size=14),
        title = element_text(size=16), legend.text = element_text(size=12))

```

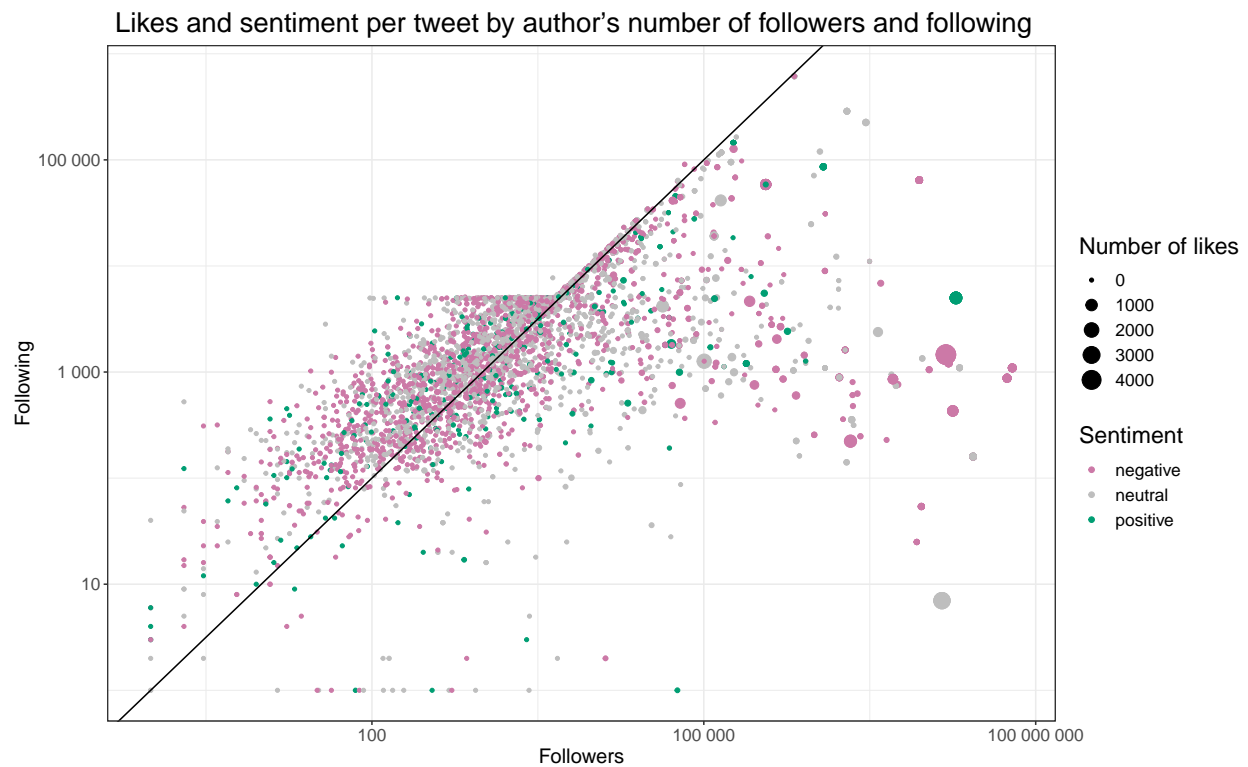


Figure 10: Bubble chart displaying the sentiment and number of likes per tweet, with the number of followers and following of the author of the tweet on the x and y axes, respectively. N.B.: This chart only displays tweets from a random sample (5%) of the complete dataset and has logarithmic scales. The follower and following counts were increased by 1 to enable conversion to a logarithmic scale. The black line shows where the author of a tweet is being followed by the same number of users as the author is following.



*This visualization shows likes and sentiment per tweet, with each dot representing a tweet. For each tweet, the number of followers of the author is given, together with the number of users the author is following. The position of the biggest dots (to the right of the black line) shows that the most liked tweets are published by users who have a large number of followers compared to the number of people they follow. For instance, the gray dot on the bottom right was published by an account with a large number of followers, but following a relatively small number of accounts. Due to the sentiment and position of the tweet, it is possible the tweet was published by a news media account of some kind.*

## 7. Claims made by people in tweets

We again include the code to obtain the data, but include the visualizations as images.

```
# we queried for
# solve.01, fight.01, address.02, change.01, show.01, reduce.01, expose.01, create.01
# and then chose interesting statements

query <- "
select distinct ?label ?role_0_val ?token_trigger_val ?role_1_val where {
  ?s wsj:onRoleSet pbddata:create.01 ;
  obio:onToken ?token_trigger ;
  wsj:withmappedrole ?role_0, ?role_1 .
  ?token_trigger rdf:value ?token_trigger_val .
  ?role_0 wsj:withpbarg pbschema:ARG0 ;
  rdf:value ?role_0_val .
  ?role_1 wsj:withpbarg pbschema:ARG1 ;
  rdf:value ?role_1_val .
  ?sent nif:word ?token_trigger .
  ?t nif:sentence ?sent ;
  obio:sentiment_label ?label .
}

query <- paste(query0, query)

# Make the GET request
response <- GET(sparql_endpoint, query = list(query = query), http::add_headers(headers))

# Check if the request was successful
if (status_code(response) == 200) {
  content_raw <- content(response, "text", encoding = "UTF-8")
  data <- fromJSON(content_raw)
  data_df <- as.data.frame(data$results$bindings)
} else {
  print(content(response, "text"))
}

df_for_printing <- data.frame(
  role1 = data_df$role_0_val$value,
  token = data_df$token_trigger_val$value,
  role2 = data_df$role_1_val$value
)
"
```

```
knitr::include_graphics("../Images/ClaimsEcon.png")
```

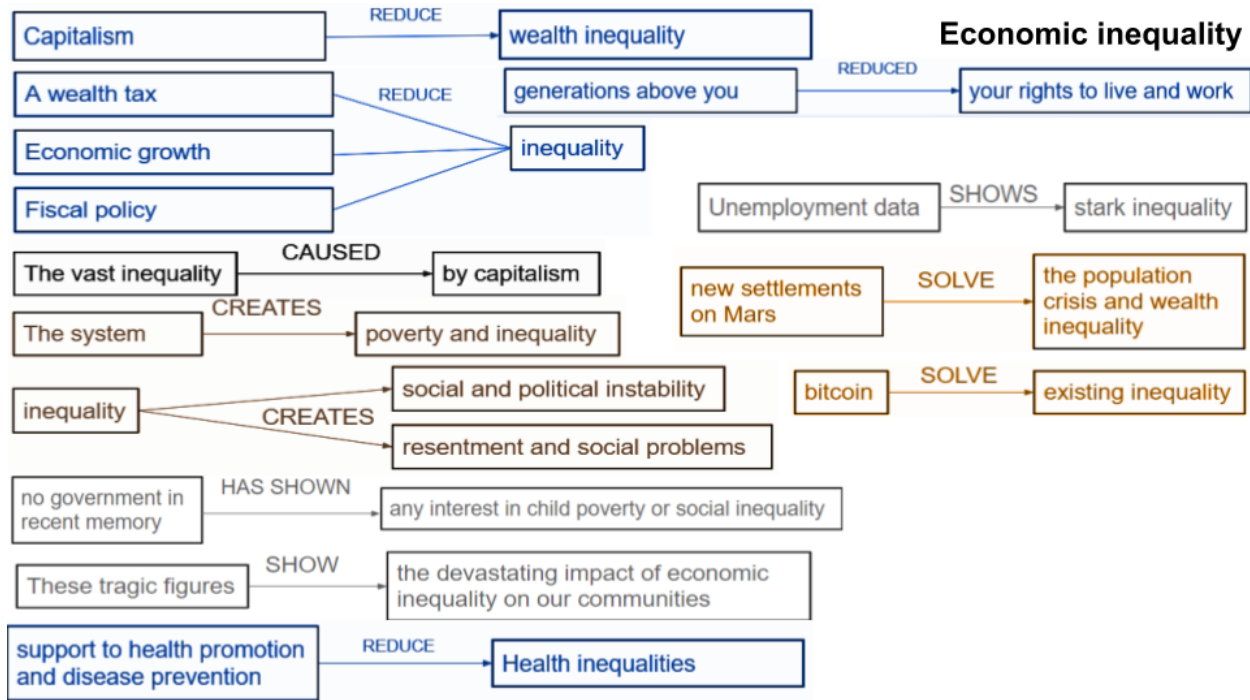


Figure 11: Claims made by Twitter users about economic inequality. N.B.: The claim extraction process does not account for negations, so some statements might not accurately reflect the content of the tweets.

When discussing economic inequality, multiple points of view are expressed by Twitter users. One user highlights how the system creates inequality, while others argue that social problems are caused by inequality. Moreover, there are conflicting views on capitalism, with one user stating that capitalism reduces wealth inequality, and the other one highlighting how vast inequalities are caused by capitalism itself. Some of the themes discussed are economic growth, fiscal policy, and unemployment data, which all connect to the theme of economic inequality. Some users state that Mars settlements, as a manifestation of human expansion beyond Earth, and the ascent of Bitcoin, a novel form of digital currency, hold promise as transformative solutions to societal inequalities.

```
knitr::include_graphics("../Images/ClaimsRacial.png")
```

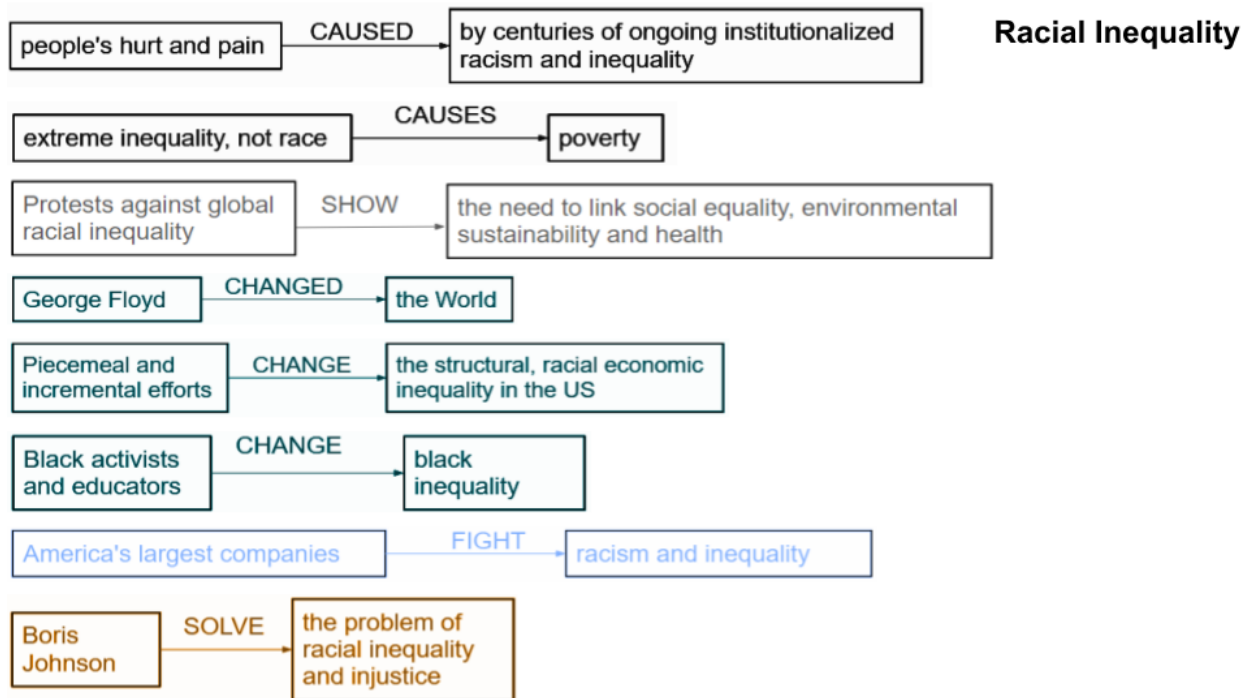


Figure 12: Claims made by Twitter users related to racial inequality.

*In tweets about racial inequality, a number of people and organizations are linked to changes and solutions to racial inequality: George Floyd, Black activists and educators, America's largest companies and Boris Johnson. Collective action is highlighted by mentioning protests and incremental efforts.*

```
knitr::include_graphics("../Images/ClaimsCovid.png")
```

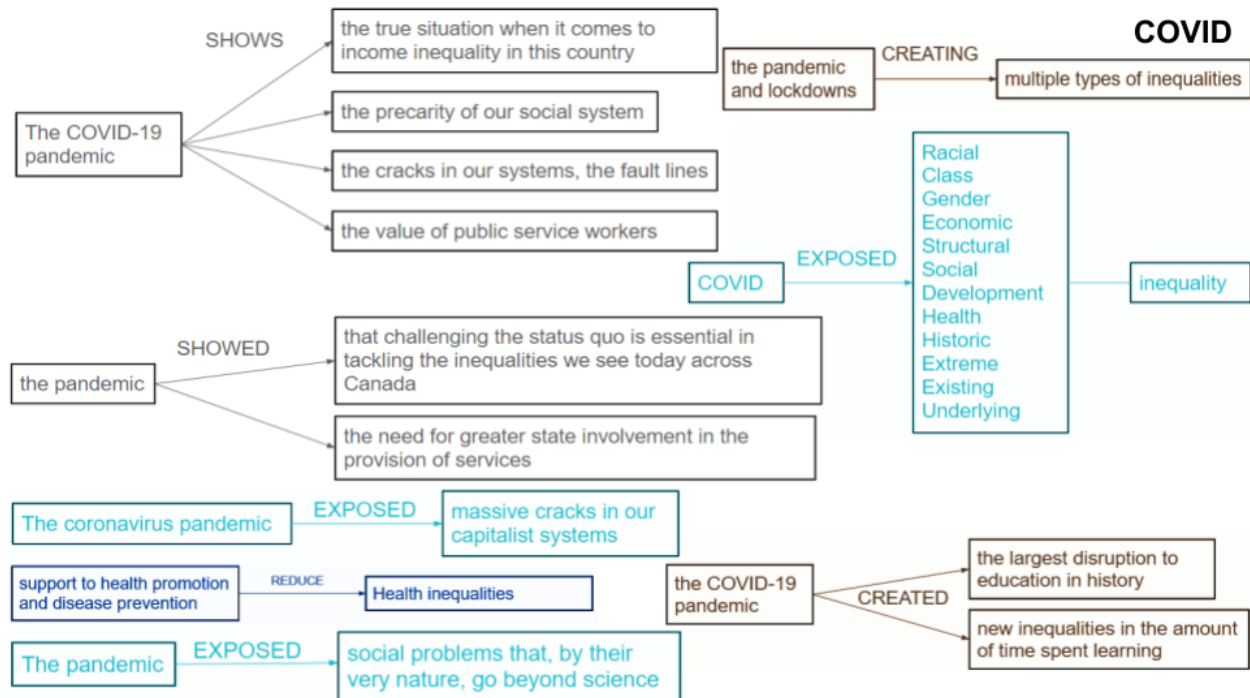


Figure 13: Claims made by Twitter users regarding COVID-19.

*When discussing COVID, multiple users mention how the pandemic exposed social problems and issues in the system, which could be governmental, economic, or educational. Some call for greater state involvement, or praise the value of public service workers, who were highly involved in the worldwide pandemic response.*

```
knitr::include_graphics("../Images/ClaimsWe.png")
```

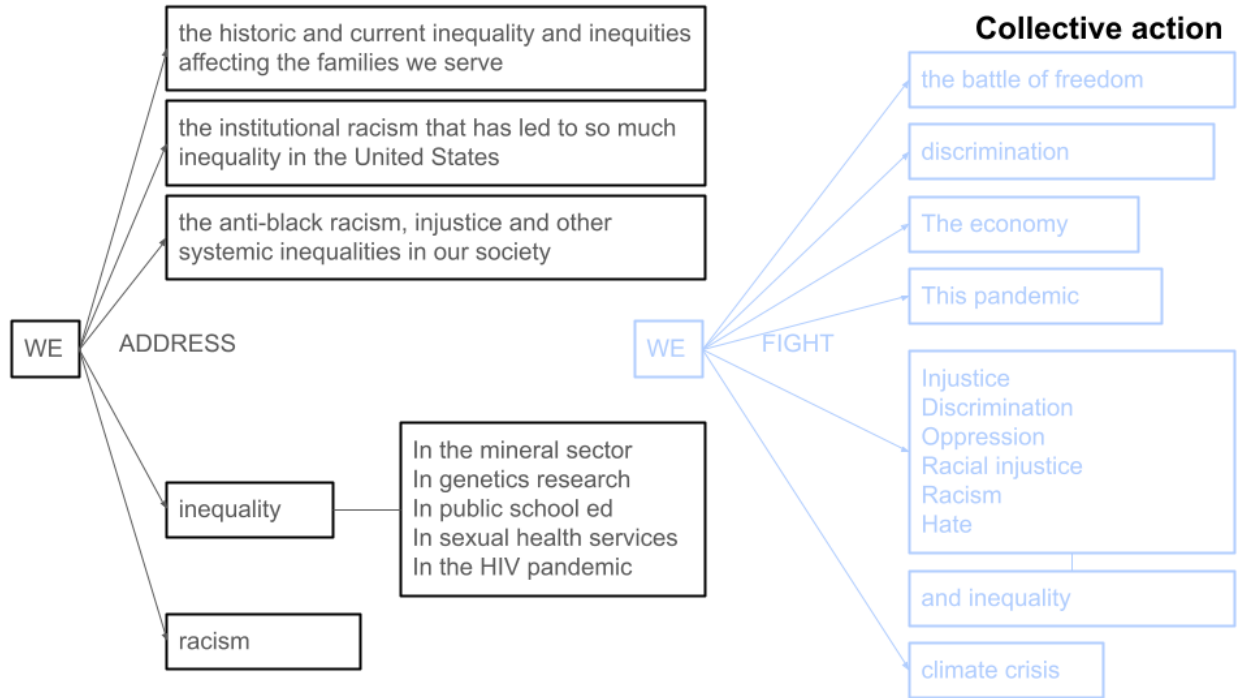


Figure 14: Claims made by Twitter users regarding collective action.

*In discussions concerning collective action, a wide range of social issues are discussed. Regarding social advocacy, the chart shows that Twitter users address inequalities affecting black people and families. They also address the presence of inequality across different spaces, such as schools, research, health services and the mineral sector.*

This work is licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Suggested Citation:

Belevi, A., Beijen, E., Borth, M., & Tacconi, C. (2024). Visualizing Social Debates on Inequality from a Large Social Media Knowledge Graph.