Comparing AI Models through Twitter Sentiment

AUTHOR
Caleb Boateng, Noah Lazar, Madeleine Schwarz

AFFILIATION
Middlebury College

▼ Code

```
library(kableExtra)
library(tidyverse)
library(dplyr)
library(lubridate)
library(tm)
library(textclean)
library(tidytext)
library(stringr)
library(syuzhet)
```

▼ Code

```
chatgpt_data <- read.csv("/Users/calebboateng/Downloads/DATA SCIENCE/chatgpt.csv",
        stringsAsFactors = FALSE)
bard_data <- read.csv("/Users/calebboateng/Downloads/DATA SCIENCE/bard.csv",
        stringsAsFactors = FALSE)
```

▼ Code

```
clean_text <- function(text) {
  text %>%
    removeNumbers() %>%
    removePunctuation() %>%
    stripWhitespace() %>%
    removeWords(stopwords("en")) %>%
    tolower()
}

bard_data$text <- sapply(bard_data$text, clean_text)
chatgpt_data$text <- sapply(chatgpt_data$text, clean_text)
```

▼ Code

```
ai_keywords <- c("artificial intelligence", "AI", "machine learning", "ChatGPT",
        "Bard")

bard_data <- bard_data %>%
  filter(str_detect(text, paste(ai_keywords, collapse = "|")))

chatgpt_data <- chatgpt_data %>%
  filter(str_detect(text, paste(ai_keywords, collapse = "|")))
```

▼ Code

```
get_average_sentiment <- function(text) {
  sentiment_scores <- get_sentiment(text, method = "syuzhet")
  mean(sentiment_scores, na.rm = TRUE)
}


bard_data$sentiment_score <- sapply(bard_data$text, get_average_sentiment)
chatgpt_data$sentiment_score <- sapply(chatgpt_data$text, get_average_sentiment)
```

Introduction

Research Question: How do public perceptions of AI chatbot models ChatGPT and Bard compare through sentiment? What are some of the frequent words that indicate the sentiment of these AI chatbots? What does this tell us about each model?

For our final project, our curiosity led us to a fascinating dataset on Kaggle, aptly named “AI-based Platforms.” Curated by Sina, the dataset offers a unique comparison between various AI models—ChatGPT, Bard, Runway, MidJourney, and Fireflies—and user-generated tweets. Our initial analysis revealed a distinct classification: ChatGPT and Bard as conversational AI, and MidJourney and Runway as AI-driven art generators. Fireflies, however, stood apart in its functionality. Intrigued by the nuances of chatbot technologies, we decided to narrow our focus to ChatGPT and Bard, aiming to uncover public sentiments towards these AI chatbots through Twitter tweets.(Tavakoli 2023)

Our journey into data processing introduced us to the practical application of functions: reusable blocks of code activated upon call. We designed a function named ‘clean_text,’ crafted to refine our dataset by removing numerals, punctuation, and redundant white spaces. It also eliminated commonly used but informationally sparse words, known as stop words (like ‘and’, ‘the’, ‘in’), and standardized all text to lowercase for uniformity. This cleanup was made seamless with the ‘tm’ package.(Feinerer and Hornik 2023)

To apply our ‘clean_text’ function specifically to the text columns of the Bard and ChatGPT datasets, we learned the use of the dollar sign syntax and the ‘apply’ function in R. Our toolkit expanded with the addition of the ‘Lubridate,’ ‘tm,’ and ‘textclean’ packages, each serving a specific purpose: ‘Lubridate’ for managing date-time data, ‘tm’ for text analysis and manipulation, and ‘textclean’ for further text refinement.[Grolemund and Wickham (2011)](Feinerer, Hornik, and Meyer 2008)(Feinerer and Hornik 2023)

While filtering our data using relevant keywords for ChatGPT and Bard, we encountered the limitations of our keyword list and the processing speed of R, hinting at potential data loss. To quantify the emotional tone of the tweets, we turned to sentiment analysis, leveraging the ‘syuzhet’ package developed by Matthew Jockers and Stanford University.This package, adept at discerning sentiment, echoes our approach to text cleaning, applying a similar methodology to gauge the positivity or negativity of keywords.(Jockers 2015)

The essence of our project lies in this sentiment analysis, which serves as a powerful tool to gauge public perception of the AI models under study. By analyzing the sentiment scores, we can ascertain the general mood of the tweets, identifying whether they lean towards a positive or negative view of ChatGPT and Bard. This analysis not only highlights the prevailing attitudes towards these chatbots but also provides valuable insights into user experiences and perceptions, painting a comprehensive picture of their impact in the realm of artificial intelligence.

Results

▼ Code

```
library(ggplot2)
library(dplyr)

bard_data$dataset <- "Bard"
chatgpt_data$dataset <- "ChatGPT"


combined_data <- bind_rows(bard_data, chatgpt_data)

ggplot(combined_data, aes(x = sentiment_score)) +
  geom_histogram(bins = 30, aes(fill = dataset), alpha = 0.5) +
  labs(x = "Sentiment Score", y = "Count", title = "Sentiment Score Distribution") +
  facet_wrap(~ dataset, scales = "free") +
  theme_minimal()
```

Graph 1: Distribtuion of Sentiment Score

For this figure, we looked at the sentiment score among users for the AI models ChatGPT and Bard to see which is the more well-liked chatbot. In this graph we can see that there is more positive sentiment in users of ChatGPT than users of Bard, and in general, users gave higher scores as well. However, in ChatGPT there is also slightly more negative sentiment. Overall, the results show that ChatGPT is the more well-liked AI model and that there are many more users with positive sentiment than users of Bard.

▼ Code

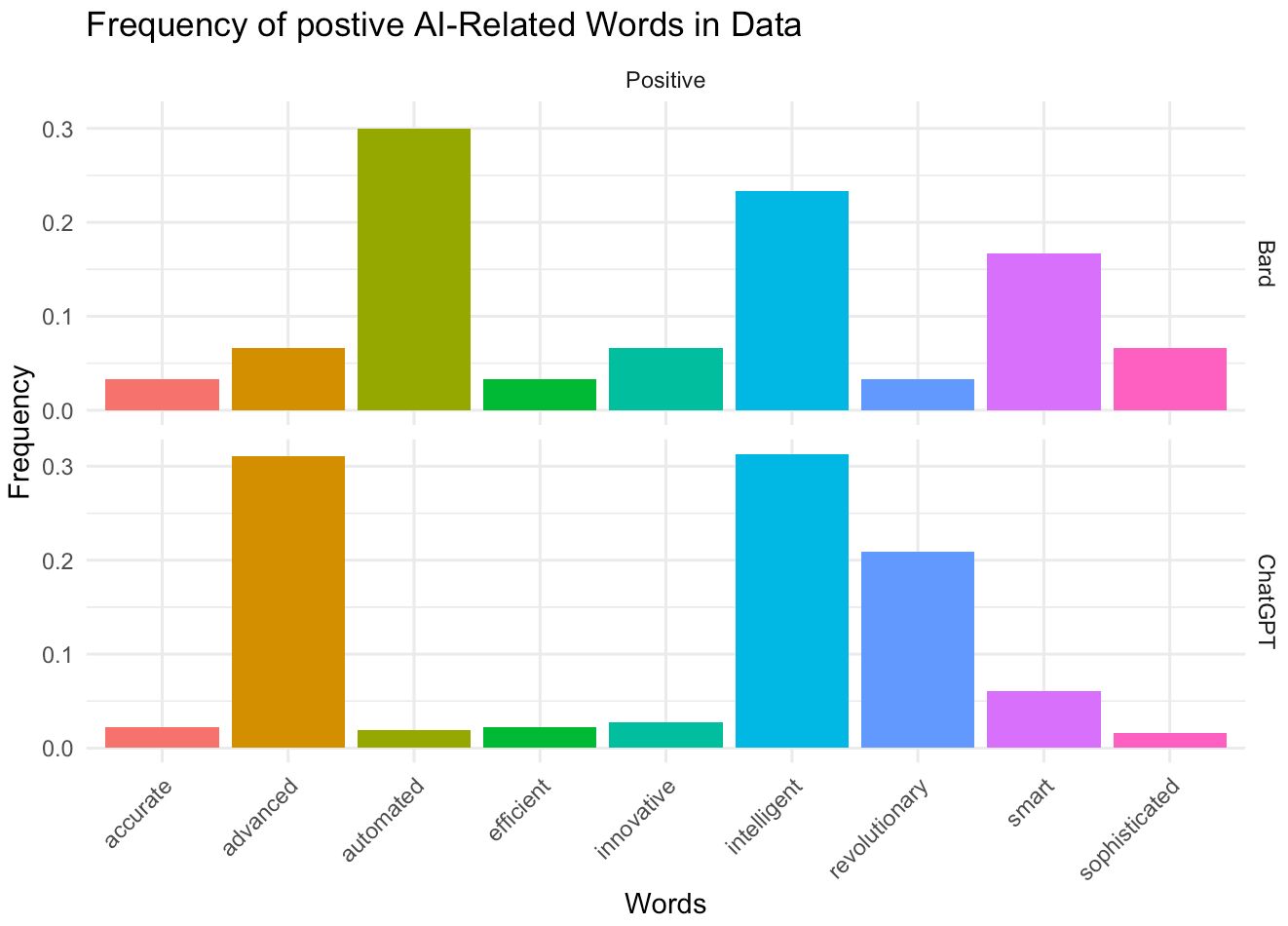
```
ai_positive_words <- c('innovative', 'intelligent', 'advanced', 'efficient',
        'revolutionary',
                       'smart', 'accurate', 'automated', 'cutting-edge',
        'sophisticated')


chatgpt_counts <- chatgpt_data %>%
  unnest_tokens(word, text) %>%
  filter(word %in% ai_positive_words) %>%
  count(word) %>%
  mutate(proportion = n / sum(n), ai = "ChatGPT")%>%
  arrange(desc(n))

bard_counts <- bard_data %>%
  unnest_tokens(word, text) %>%
```

```
  filter(word %in% ai_positive_words) %>%
  count(word) %>%
  mutate(proportion = n / sum(n), ai = "Bard")%>%
  arrange(desc(n))


rbind(chatgpt_counts %>% mutate(ai = "ChatGPT", sentiment = "Positive"),
      bard_counts %>% mutate(ai = "Bard", sentiment = "Positive"))%>%
  ggplot(aes(x = word, y = proportion, fill = word)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Frequency of postive AI-Related Words in Data",
       x = "Words", y = "Frequency") +
  theme(legend.position = "none", axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_grid(ai~sentiment)
```



Graph 2: Frequency of Positive AI related words

Bard is frequently associated with being “innovative” and “revolutionary,” which could suggest that users view it as a cutting-edge and transformative technology. ChatGPT, on the other hand, scores higher on being perceived as “intelligent” and “smart,” indicating that users may find it to exhibit strong cognitive abilities and practical intelligence. Both AI models are similarly regarded in terms of being “advanced” and “automated,” reflecting a recognition of their technological sophistication. In addition, the term “sophisticated” is more associated with Bard, possibly implying that it is seen as

technologically refined or complex. The relatively low frequency of “accurate” and “efficient” for both models could point to these attributes being less pronounced in user discussions or less differentiated between the two AIs.

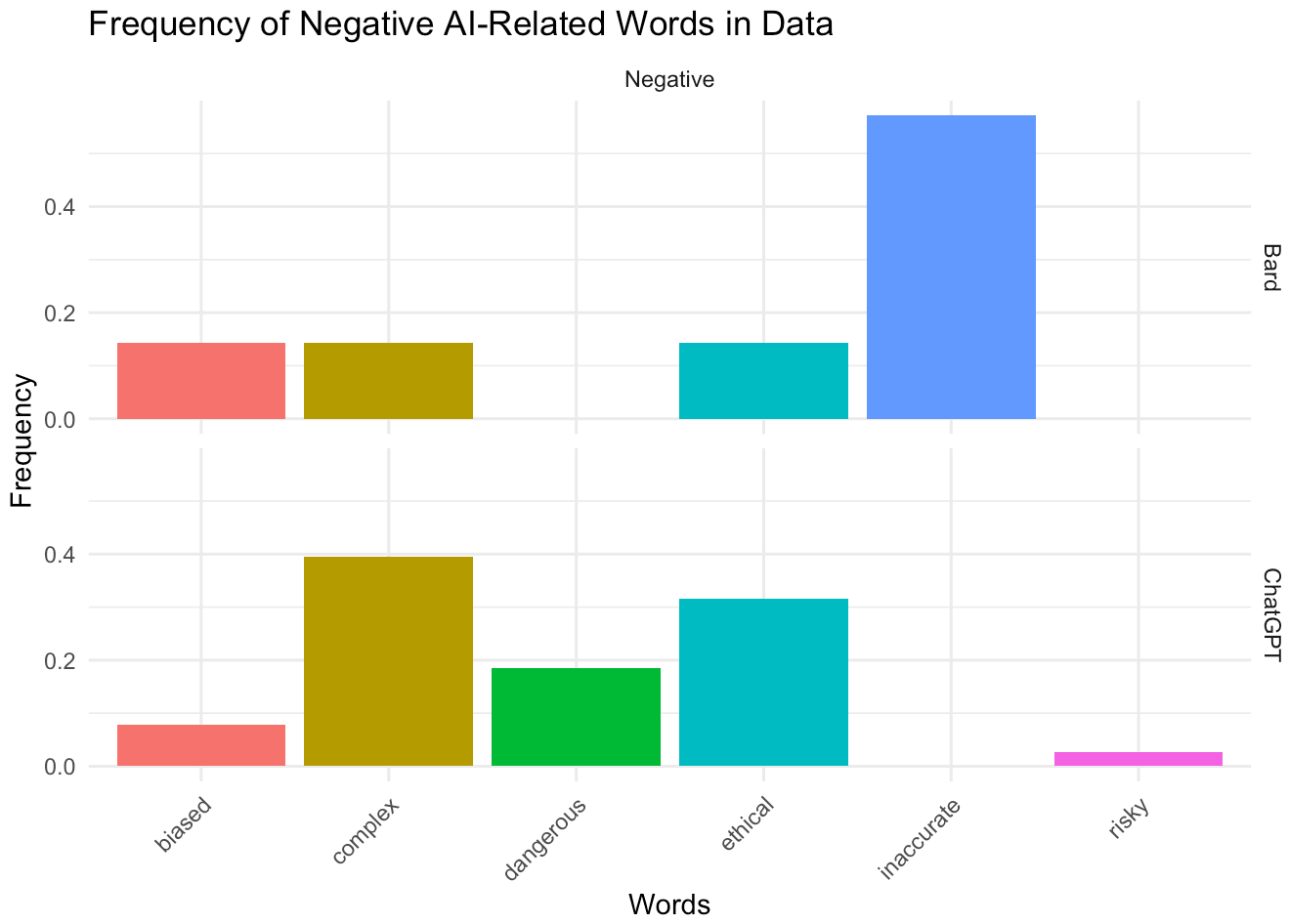
This data allows us to infer that both ChatGPT and Bard are both highly acknowledged for their technological skills, but they each have their own distinct attributes that resonate differently with public.

▼ Code

```
ai_negative_words <- c('risky', 'biased', 'unreliable', 'complex', 'dangerous',
                       'intrusive', 'impersonal', 'inaccurate', 'ethical', 'expensive')
chatgpt_negative_counts <- chatgpt_data %>%
  unnest_tokens(word, text) %>%
  filter(word %in% ai_negative_words) %>%
  count(word) %>%
  mutate(proportion = n / sum(n), ai = "Chatgpt")%>%
  arrange(desc(n))

bard_negative_counts <- bard_data %>%
  unnest_tokens(word, text) %>%
  filter(word %in% ai_negative_words) %>%
  count(word) %>%
  mutate(proportion = n / sum(n), ai = "Bard")%>%
  arrange(desc(n))

rbind(chatgpt_negative_counts %>% mutate(ai = "ChatGPT", sentiment = "Negative"),
      bard_negative_counts %>% mutate(ai = "Bard", sentiment = "Negative"))%>%
  ggplot(aes(x = word, y = proportion, fill = word)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Frequency of Negative AI-Related Words in Data",
       x = "Words", y = "Frequency") +
  theme(legend.position = "none", axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_grid(ai~sentiment)
```



Graph 3: Frequency of Negative AI related words

The bar graph illustrates the frequency of negative keywords that summarize sentiment toward these AI models, and it is evident that both ChatGPT and Bard are perceived as “biased” to a similar extent, reflecting a shared concern among users about potential impartiality within AI interactions. Bard, notably, is more frequently associated with being “complex” and “expensive,” which could indicate a user perception of Bard as a more intricate and cost-intensive option. Conversely, ChatGPT is more often described as “impersonal” and “inaccurate,” hinting at a possible gap in fulfilling user expectations of personalization and precision. Interestingly, “unreliable” is not mentioned for both chatbots, suggesting a general user trust in their performance stability. Other terms like “intrusive,” “dangerous,” and “risky” are also linked to these models, possibly indicating that these are not predominant concerns.

In comparison, to Graph 2, there are far less negative sentiment words mentioned than positive sentiment words. We can conclude that overall ChatGPT and Bard have a positive perception throughout the twitter tweets.

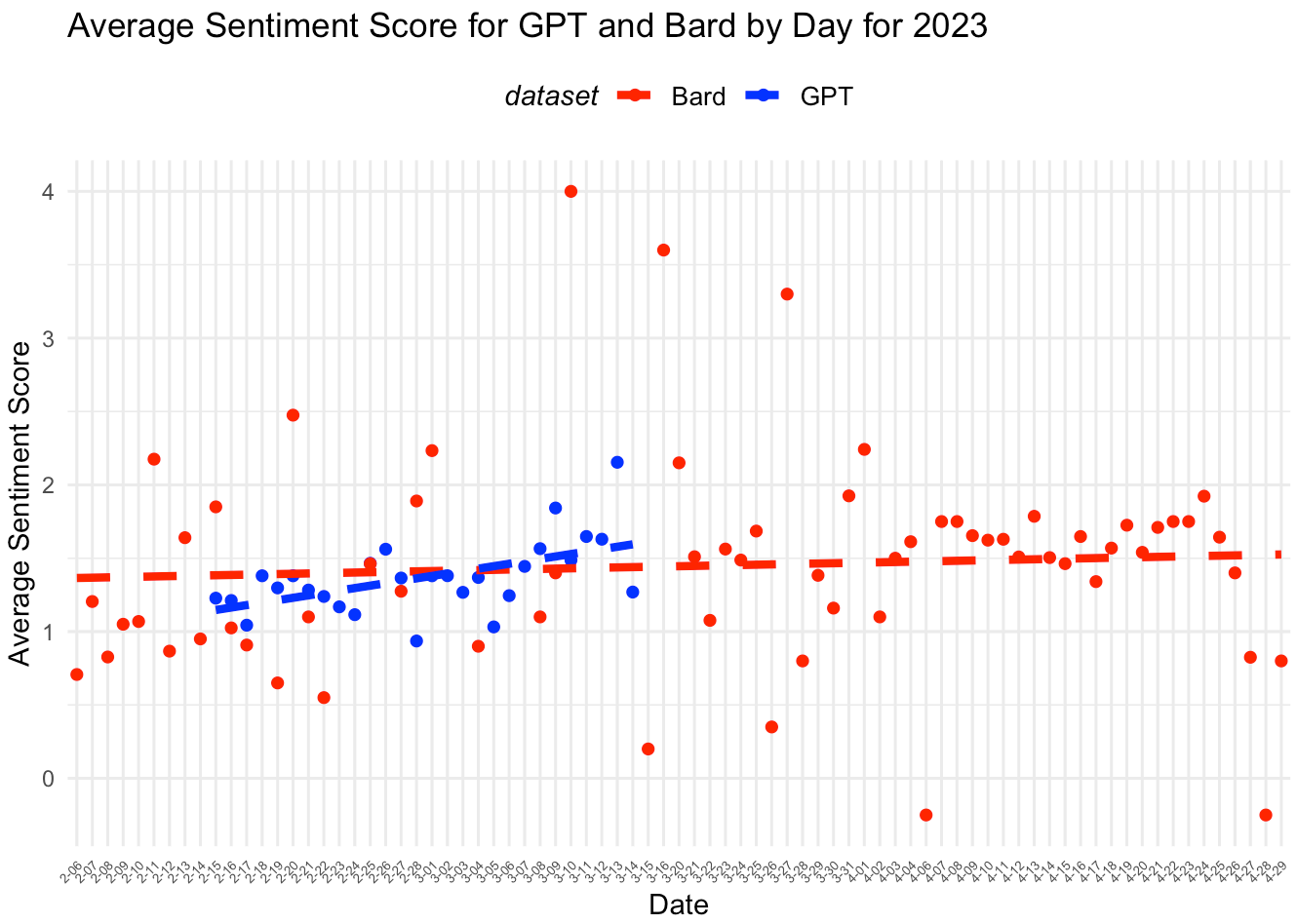
▼ Code

```
gpt_ave <- chatgpt_data %>%
  group_by(created_at) %>%
  mutate(created_at = str_sub(created_at, start = 7, end = 10))%>%
  summarize(avg_sentiment = mean(sentiment_score, na.rm = TRUE))
```

```r
bard_ave <- bard_data %>%
  group_by(created_at) %>%
  mutate(created_at = str_sub(created_at, start = 7, end = 10))%>%
  summarize(avg_sentiment = mean(sentiment_score, na.rm = TRUE))


gpt_bard <- bind_rows(
  mutate(gpt_ave, dataset = "GPT"),
  mutate(bard_ave, dataset = "Bard")
)


ggplot(gpt_bard, aes(x = created_at, y = avg_sentiment, color = dataset)) +
  geom_point(shape = 16, size = 2) +
  geom_smooth(method = "lm", se = FALSE, aes(group = dataset, color = dataset),
              linetype = "dashed", size = 1.5) +
  labs(
    x = "Date",
    y = "Average Sentiment Score",
    title = "Average Sentiment Score for GPT and Bard by Day for 2023"
  ) +
  scale_color_manual(values = c("GPT" = "blue", "Bard" = "red")) +
  theme_minimal() +
  theme(
    legend.position = "top",
    legend.title = element_text(face = "italic"),
    legend.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, size = 5, hjust = 1)
  )
```



Graph 4: Average Sentiment Score by Day

For this figure, we took the average sentiment score for that day. This showed us how the models were improving or not improving. We can also see the data for ChatGPT spans less of a period, but the trend is increasing in sentiment score. In the Bard data set the data spans a longer period, but more spread out with a trend that seems extremely linear. AI models learn from themselves, and this shows that ChatGPT is learning and adapting to users. Chatgpt is on a clear positive slope, improving daily. If this trend was to continue we may see Chatgpt have a higher sentiment score.

▼ Code

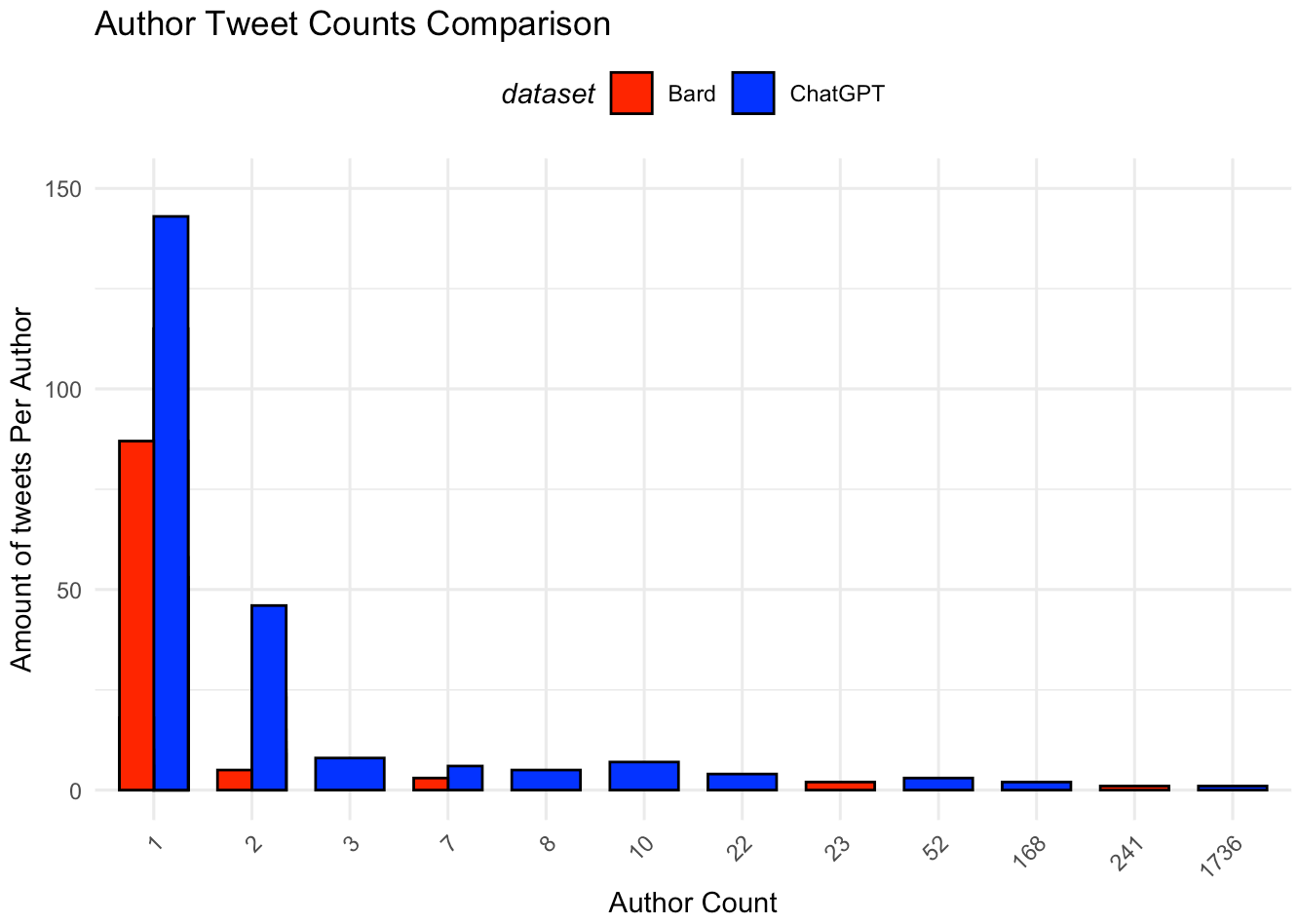
```
chatgpt_counts <- chatgpt_data %>%
  group_by(author_id) %>%
  summarise(amount_tweet = n()) %>%
  group_by(amount_tweet) %>%
  summarise(amount_author = n())

bard_counts <- bard_data %>%
  group_by(author_id) %>%
  summarise(amount_tweet = n()) %>%
  group_by(amount_tweet) %>%
  summarise(amount_author = n())

combined_counts <- bind_rows(
```

```
  mutate(chatgpt_counts, dataset = "ChatGPT"),
  mutate(bard_counts, dataset = "Bard")
)

ggplot(combined_counts, aes(x = reorder(factor(amount_author), amount_author), y =
       amount_tweet, fill = dataset)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7, color = "black") +
  labs(x = "Author Count", y = "Amount of tweets Per Author", title = "Author Tweet
       Counts Comparison") +
  scale_fill_manual(values = c("ChatGPT" = "blue", "Bard" = "red")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
       legend.position = "top",
       legend.title = element_text(face = "italic")) +
  scale_x_discrete(labels = function(x) stringr::str_wrap(x, width = 10)) +
  ylim(0, 150)
```



Graph 5: Number of Tweets

For this figure, we looked at the authors and tweets produced per author. This shows us in the data set that some authors are tweeting all the time and adding only their perspective to the data set. This shows us that this data can be biased because one user's input affects the sentiment score negatively. Although we can see that some users imputed a lot of data, there are so many tweets the effect that some of these users have is minimal.

Conclusion

We see a higher sentiment score for Bard, which means, users think the bot is better than ChatGPT. This score is very similar to ChatGPT with it only being .07 less, but this is still a significant difference. This shows that when the data was taken ChatGpt was worse than Bard. Bard had better reviews on this data-set, so therefore for this project we conclude that Bard is better. As an AI-Model that is always evolving this can change.

Discussion

In future analysis, we could explore more present data because AI models change very quickly and user's opinions have changed over time about AI. This would be interesting to look at every time a new update to the software was rolled out. I would love to look at more tangible things like computing power of the AI-Model or amount of user inputs daily. Also if we had more computational power that would allow us to search for more keywords at once for the top 10 words graph, but R was too Slow.

Author Contributions

Caleb and Noah started this project over break, finding the data set we wanted to work on and then thinking through the sentiment score. This allowed us to add value to words. From here Madeline joined our group. We all then worked on graphs together, but we all had individual ideas and needed support figuring out the code at the time. We all created either one or two graphs and created the write-up for that as well. We tried to divide work equally. This gave us the opportunity to change and check of others work so everything was correct.

Acknowledgments

We want to thank professor Emily and Louise for their continued support for this whole term and always answering our questions in and out of class. We also want to thank professor Lyford for his support creating better figures and always being willing to help.

References

Feinerer, Ingo, and Kurt Hornik. 2023. *Tm: Text Mining Package*. https://tm.r-forge.r-project.org/.

Feinerer, Ingo, Kurt Hornik, and David Meyer. 2008. "Text Mining Infrastructure in r." *Journal of Statistical Software* 25 (5): 1–54. https://doi.org/10.18637/jss.v025.i05.

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. https://www.jstatsoft.org/v40/i03/.

Jockers, Matthew L. 2015. *Syuzhet: Extract Sentiment and Plot Arcs from Text*. https://github.com/mjockers/syuzhet.

Tavakoli, Sina. 2023. "AI Based Platforms." *Kaggle*. https://www.kaggle.com/datasets/sinatavakoli/ai-based-platforms.