Text

Description automatically generatedA Sentiment Analysis of ChatGPT

By Diane Sullivan

Problem Statement

ChatGPT (Chat Generative Pre-trained Transformer) developed by OpenAI, is an artificial intelligence chatbot based on large language models. In the short time that this chatbot has been publicly available, I have heard mixed reactions to this tool. Teachers at my children’s schools are worried that ChatGPT is an effective and very accessible tool that will be used for cheating. My friends are openly wondering if they might be professionally at risk to be replaced by ChatGPT. My career counselor at UNCW encouraged me to use ChatGPT to assist with writing my resume and cover letters for job applications. She felt it was an excellent tool for streamlining the application process. I feel confused by the varying sentiments toward this new chatbot. I would like to know the feeling in the general public regarding the advance of ChatGPT. Since social media platforms generate big data through user posts, it is possible to gauge mass attitude towards a specific topic or product by examining the social media. (1) Therefore, I will use Twitter as my social media source to gauge public sentiment.

* I will investigate whether people have more positive or negative sentiment towards ChatGPT as gauged by ChatGPT themed posts on Twitter.

Literature Review

OpenAI launched ChatGPT in November 2022 and then in February 2023 it upgraded the chatbot to the turbo 3.5 GPT.. In February of 2023, Microsoft announced that it was investing $10 billion in OpenAI. Microsoft is in this conversation with large language model chatbots with their Bing bot. The usability of chatbots in their recent iterations, points to the idea that chatbots are here to stay. “There is no question that [generative AI] is going to be used—it’s not just a novelty,” says David Autor, an MIT labor economist and a leading expert on the impact of technology on jobs. (2) I will look specifically at ChatGPT. My literature review of the powers and prowess of ChatGPT indicate that this chatbot is a mixed bag!

ChatGPT can unlock creativity. Of course, it can mimic or flat out copy others’ creative process, but it can also spur on the creative process. Perhaps ChatGPT is the friend that gets you through writers’ block! ChatGPT can offer personalized tutoring. It can also better prepare students to work alongside A.I. systems as adults, not be replaced by ChatGPT, but work alongside it. This chatbot can write you a song or an essay, it can write computer code, and of course it can solve math problems. Like all resources, it can be used for plagiarizing. ChatGPT sometimes gives wrong answers to factual questions. It may also give a different answer when asked the same question more than once. (3)

Economists are uncertain how this revolutionary chatbot will change employment numbers or boost workers’ productivity. The optimistic sentiment is that since this chatbot requires no coding and has easy interface, it may improve expertise and capabilities amongst workers across all industry sectors and give a big economic boost to the companies engaging ChatGPT. The pessimistic view is that a few elite tech companies will harness this power to make large profits while putting humans out of business. Careers that require creativity and logic, formerly thought to be automation proof, will be replaced by Chat-GPT. Those are often more lucrative careers. If companies replace relatively well-paying white-collar jobs with chatbots, workers will be left with lower-paying service employment while the few who are best able to exploit the new technology reap all the benefits.  The economic worry around ChatGPT is not about unemployment but more about under employment. (2).

Another sentiment analysis on Twitter data from December 2022-Janauary 2023 by Adem Korkmaz and Cemal Aktürk found that the majority of tweets they analyzed had positive emotions. Of note, this is not the turbo 3.5 GPT that is now publicly available but rather the former generation of Chat GPT. It was seen in their analysis that a small portion of the users got erroneous-false results with ChatGPT queries and they were not satisfied with this situation. However, the majority of tweets identified being satisfied with using this AI chatbot and they considered the experience successful. (1)

Jessica Bethea completed a sentiment analysis regarding ChatGPT using a twitter dataset from February of 2023. She found positive sentiment in her dataset in 3 of the four lectionaries she analyzed. Her Bing lexionary matched with slightly more negative than positive words in a sentiment review. However, the other 3 lectionaries Ms. Bethea analyzed (AFINN, Loughran, nrc) all had more positive sentiment. (4)

Going in chronological order, the next dataset is the timeframe I analyzed: April 3 – April 18, 2023 ChatGPT-themed tweets.

Data

I will complete this sentiment analysis in R Studio.

### Load the required libraries and read in the data

#install.packages('tm')  
#install.packages('quanteda')  
library(tidyverse)  
library(tidytext)  
library (tm)  
library (quanteda)  
library(textdata)  
library(gridExtra) #viewing multiple plots together

#install.packages("wordcloud2")  
library(wordcloud2)

## Warning: package 'wordcloud2' was built under R version 4.2.3

#install.packages("RColorBrewer")  
library(RColorBrewer)

chatgpt\_original <- read\_csv("chatgpt\_kaggle.csv")

This dataset if from Kaggle: [#ChatGPT 1000 Daily 🐦 Tweets | Kaggle](https://www.kaggle.com/datasets/edomingo/chatgpt-1000-daily-tweets) and is owned by Enric Domingo.

This dataset is updated everyday on Kaggle with tweets that contain the phrases: **"ChatGPT", "GPT3", or "GPT4".** I downloaded the dataset on April 18, so my data covers the period 4/3/2023 – 4/17/2023. This dataset collected has 17,003 observations and 17 variables.

Here are the 17 variables in this dataset:

Tweet id: unique identifier for each tweet

Tweet created: date and time of tweet creation (timestamp)

Text: Tweet context, this is the sentiment I will analyze

Language: the tweets text language

User ID: Twitter’s unique user ID

User Name: the author’s public name on Twitter

User UserName: the author’s twitter account username, @handle, for example

User Location: the author’s public location

User Description: the author’s public profile bio

Users follower count: the number of followers of the author’s account

User following count: the number of followed accounts from the author’s account

User tweet count: the number of tweets the author has published

Retweet count: number of times the tweet was retweeted

Source: device or app used to publish the tweet

User verified: marked “true” if the user has a blue check mark, known as verified

like count: number of likes to the tweet

reply count: number of reply messages to the tweet

impression count: number of times the tweet has been seen

Start cleaning and pre-processing this dataset.

We can use the function filter() to delete texts that are in languages other than English:

English <-chatgpt\_original %>%   
 filter(lang=="en")  
dim(English)

## [1] 7843 19

Now the dataframe contains only tweets in English language: 7843 observations with the above listed 19 variables.

I will use the function select() to rename the columns and reduce the variables. I’ll change the name of the column tweet\_id to id. I will use the following 6 variables throughout my analysis: text, tweet id, retweet count, tweet creation timestamp, user name, user tweet count).

tweets <-English %>%   
 select (text,id=tweet\_id,retweet\_count,tweet\_created,user\_name,user\_tweet\_count)  
dim(tweets)

## [1] 7843 6

Remove contractions by creating a function that # handles most scenarios using gsub(), and then apply # that function across all tweets.

fix.contractions <- function(doc) {  
 # "won't" is a special case as it does not expand to "wo not"  
 doc <- gsub("won't", "will not", doc)  
 doc <- gsub("can't", "can not", doc)  
 doc <- gsub("n't", " not", doc)  
 doc <- gsub("'ll", " will", doc)  
 doc <- gsub("'re", " are", doc)  
 doc <- gsub("'ve", " have", doc)  
 doc <- gsub("'m", " am", doc)  
 doc <- gsub("'d", " would", doc)  
 doc <- gsub("@\\w+", "", doc) #removes @handles  
 # 's could be 'is' or could be possessive: it has no expansion  
 doc <- gsub("'s", "", doc)  
 return(doc)  
}  
  
# fix (expand) contractions  
tweets$text <- sapply(tweets$text, fix.contractions)

## Data Analysis

## Word Frequency

#define some colors to use throughout  
my\_colors <- c("#E69F00", "#56B4E9", "#009E73", "#CC79A7", "#D55E00", "#D65E00")  
  
  
tidy\_tweets %>%  
 count(word, sort = TRUE) %>% #get the n top words from the tidied, clean, filtered dataset using count() and top\_n()   
 top\_n(15) %>%  
 ungroup() %>%   
 mutate(word = reorder(word, n)) %>% #sort words according to the count using reorder()and reassign the ordered value to word using mutate()  
 ggplot() +  
 geom\_col(aes(word, n), fill = my\_colors[2]) +  
 xlab("") +   
 ylab("Word Count") +  
 ggtitle("Most Frequently Used Words in Chatgpt Related Tweets") +  
 coord\_flip()

## Selecting by n

Chart

Description automatically generated

Using the tidy format, I did a simple evaluation and visualization of the most frequently used words in the full set of tweets. With the word frequency, you can identify the words common in all of the tweets but it’s not particularly meaningful: “tools” occurred most. The word tools occurred over 500 times; ChatGPT being a new and powerful tool, the frequent use of the word “tool” makes sense. Same for the fourth most occurring word: “OpenAI”. But how do people feel about it? What is the sentiment, positive or negative? I don’t really know yet.

Interestingly the word “tip” appears 187 times in the dataset and “iceberg” appears 180. I infer that the phrase “tip of the iceberg” seems to catch common feeling that we are at the beginning of a new phase with much more to come in developments with artificial intelligence. That observation is consistent with my literature review findings.

* Term Frequency (tf): Number of times a term occurs in a document
* Document Frequency (df): Number of documents that contain each word

tidy\_text\_tfidf<- tweets %>%  
 unnest\_tokens("word", text) %>%  
 anti\_join(stop\_words) %>%  
 count(user\_tweet\_count, word) %>%  
 bind\_tf\_idf(word, user\_tweet\_count, n)

tidy\_text\_tfidf

## # A tibble: 78,916 × 6  
## user\_tweet\_count word n tf idf tf\_idf  
## <dbl> <chr> <int> <dbl> <dbl> <dbl>  
## 1 1 13333 1 0.0122 7.93 0.0967  
## 2 1 28 1 0.0122 6.43 0.0784  
## 3 1 4x 1 0.0122 4.14 0.0505  
## 4 1 500 1 0.0122 7.24 0.0883  
## 5 1 55,000 1 0.0122 4.14 0.0505  
## 6 1 ai 2 0.0244 1.23 0.0299  
## 7 1 alibaba 1 0.0122 5.37 0.0655  
## 8 1 arb 1 0.0122 4.14 0.0505  
## 9 1 arbitrum 1 0.0122 4.14 0.0505  
## 10 1 artificial 1 0.0122 4.24 0.0518  
## # … with 78,906 more rows

# TF-IDF scores close to the value of 1 point to words with more weight or relevance in the text; words with lower TF-IDF numbers, closer to zero, are of less importance in the text. It is also likely that a term that occurs fewer times in a text, or shows up in many documents, has less importance, a lower TF-IDF. If a term is universally present in many texts, it will have the lowest TF-IDF score, thus indicating its lack of importance.

I would use TF-IDF instead of term frequency because TF simply identifies occurrences of terms or words but does not reflect their importance. TF is a simple word count within a defined document without additional context with which to measure relevance or importance. The whole picture is revealed when adding the IDF metric which also considers the number of documents that contain the term. IDF of a rare term will most likely be a high number, whereas IDF of a frequent term is likely to be low. TF-IDF approaches zero as terms have less relevance. Thus, you could have a high TF but a low TF-IDF and immediately realize the word is overly used or universally present and has no weight or importance.

Next, I will continue sentiment analysis with three varying lexicons: bing, afinn, nrc.

#Explore the content of the lexicons

bing<-get\_sentiments("bing")  
nrc<-get\_sentiments("nrc")  
afinn<-get\_sentiments("afinn")  
  
#convert the values in the afinn lexicon to positive and negative sentiments  
afinn\_neg\_pos <- afinn %>%  
 mutate( sentiment = ifelse( value >= 0, "positive",  
 ifelse( value < 0,  
 "negative", value)))  
afinn\_neg\_pos <-afinn\_neg\_pos %>%  
 select(word, sentiment)  
  
  
#Combine the three lexicons  
sentiments <-bind\_rows(list(bing=bing,nrc=nrc,afinn=afinn\_neg\_pos),.id = "lexicon")  
  
  
new\_sentiments <- sentiments %>%  
 group\_by(lexicon) %>%  
 mutate(words\_in\_lexicon = n\_distinct(word)) %>%  
 ungroup()  
  
#Let’s look at words count per lexicons.  
new\_sentiments %>%  
 group\_by(lexicon, sentiment, words\_in\_lexicon) %>%  
 summarise(distinct\_words = n\_distinct(word)) %>%  
 ungroup() %>%  
 spread(sentiment, distinct\_words)

## `summarise()` has grouped output by 'lexicon', 'sentiment'. You can override  
## using the `.groups` argument.

## # A tibble: 3 × 12  
## lexicon words\_in\_l…¹ anger antic…² disgust fear joy negat…³ posit…⁴ sadness  
## <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int>  
## 1 afinn 2477 NA NA NA NA NA 1598 879 NA  
## 2 bing 6783 NA NA NA NA NA 4781 2005 NA  
## 3 nrc 6453 1245 837 1056 1474 687 3316 2308 1187  
## # … with 2 more variables: surprise <int>, trust <int>, and abbreviated  
## # variable names ¹​words\_in\_lexicon, ²​anticipation, ³​negative, ⁴​positive

All three lexicons have more negative than positive words.

tidy\_tweets %>%  
 mutate(words\_in\_tweets = n\_distinct(word)) %>%  
 inner\_join(new\_sentiments) %>%  
 group\_by(lexicon,words\_in\_tweets, words\_in\_lexicon) %>%  
 summarise(lex\_match\_words = n\_distinct(word)) %>%  
 ungroup() %>%  
 mutate(total\_match\_words = sum(lex\_match\_words), #Not used but good to have  
 match\_ratio = lex\_match\_words / words\_in\_tweets) %>%  
 select(lexicon, lex\_match\_words, words\_in\_tweets, match\_ratio)

## Joining, by = "word"  
## `summarise()` has grouped output by 'lexicon', 'words\_in\_tweets'. You can  
## override using the `.groups` argument.

## # A tibble: 3 × 4  
## lexicon lex\_match\_words words\_in\_tweets match\_ratio  
## <chr> <int> <int> <dbl>  
## 1 afinn 785 12015 0.0653  
## 2 bing 1134 12015 0.0944  
## 3 nrc 1589 12015 0.132

NRC seems to have the best match ratio for this twitter derived ChatGPT dataset followed by bing and then afinn. This is a low match ratio but many words don’t carry sentiment, thus they are considered neutral and will not give weight to the match ratio. Also, these lexicons were not created or validated specifically for twitter assessments, so the match ratio remains low.

Apply these sentiment dictionaries to the tidied dataset. I will start by creating twitter users’ chapgpt sentiment datasets for each of the lexicons by performing an inner\_join() on the get\_sentiments() function. Pass the name of the lexicon for each call. For this exercise, use Bing for binary sentiments, NRC for categorical sentiments, and afinn for a range of sentiments from -5 to 5.

Implement sentiment analysis using the inner join function and the “nrc” lexicon by performing an inner\_join() on the get\_sentiments() function.

tweets\_nrc <- tidy\_tweets %>%  
 inner\_join(get\_sentiments("nrc"))

## Joining, by = "word"

tweets\_afinn <- tidy\_tweets %>%  
 inner\_join(get\_sentiments("afinn"))

## Joining, by = "word"

tweets\_bing <- tidy\_tweets %>%  
 inner\_join(get\_sentiments("bing"))

## Joining, by = "word"

# Count by word and sentiment with bing

tweets\_bing %>%  
inner\_join(get\_sentiments("bing")) %>%  
count(word, sentiment, sort = TRUE) %>%  
 group\_by(sentiment) %>%  
 top\_n(10) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 # Set up the plot with aes() using the ggplot() and geom\_col(). set the graph aes so x is word and y is n and the columns are filled with sentiment.   
 ggplot(aes(word,n, fill=sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ sentiment, ncol =5, scales = "free")+  
 coord\_flip()

## Joining, by = c("word", "sentiment")  
## Selecting by n

Chart, bar chart, funnel chart

Description automatically generated

Bing lexionary: Strong negative words with over 75 unique occurances: fake, miss, rival. False and cheat also present nearly 50 times. But, even more positive words with free showing up over 225 times, followed by prompt, excited, intelligence, powerful, amazing.

summary\_bing <- tweets\_bing %>%   
 count(sentiment, sort = TRUE) %>%   
 spread(sentiment, n) %>%  
 mutate(sentiment = positive - negative) %>%  
 mutate(lexicon = "bing") %>%  
 relocate(lexicon)  
  
summary\_bing

## # A tibble: 1 × 4  
## lexicon negative positive sentiment  
## <chr> <int> <int> <int>  
## 1 bing 3128 4021 893

When I delete all words less than 3 characters, the sentiment in bing has 914 more positive sentiments than negative in the dataset of 6732 total observations. However, when I delete all words less than 2 characters, the sentiment is still positive, but less so, with 893 more positive than negative words. This is just a cautionary note; as an analyst, it is helpful to be neutral in opinion until completing a thorough look at data but how the data is pre-processed can really change the result.

tweets\_nrc %>%  
inner\_join(get\_sentiments("nrc")) %>%  
count(word, sentiment, sort = TRUE) %>%  
 group\_by(sentiment) %>%  
 top\_n(10) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 # Set up the plot with aes() using the ggplot() and geom\_col(). set the graph aes so x is word and y is n and the columns are filled with sentiment.   
 ggplot(aes(word,n, fill=sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ sentiment, ncol =4, scales = "free")+  
 coord\_flip()

## Joining, by = c("word", "sentiment")  
## Selecting by n

Timeline

Description automatically generated

“Powerful” is present in many categories: anger, anticipation, disgust, fear, joy, but in general, the word counts are higher in the categories with positive sentiments: anticipation, positive, trust, surprise, joy (that list is organized in descending order). There is negative sentiment here, for example, “fake” occurs over 125 times and “bad” almost 100 times. However, the positive sentiments outnumber the negative in word counts by categories.

tweets\_afinn %>%  
inner\_join(get\_sentiments("afinn")) %>%  
count(word, value, sort = TRUE) %>%  
 group\_by(value) %>%  
 top\_n(10) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 # Set up the plot with aes() using the ggplot() and geom\_col(). set the graph aes so x is word and y is n and the columns are filled with sentiment.   
 ggplot(aes(word,n, fill=value)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ value, ncol =5, scales = "free")+  
 theme(axis.text.x = element\_text(face = "bold", color = "blue",  
size = 8, angle = 90))+  
 coord\_flip()

## Joining, by = c("word", "value")  
## Selecting by n

Timeline

Description automatically generated

In this graph of the afinn categories, again, positive sentiment, shown here in all categories with values above 0 have much higher word counts than categories with negative sentiment.

A picture containing map

Description automatically generated

Findings

In April 2023, the sentiment around ChatGPT was more positive than negative based on tweets in that timeframe. The highest word counts from the tweets are neutral words, like “tools”,” people”, “openai”. But definite positive sentiment is seen in the lexicon matching.

Looking at the tweets in the context of three lexicons that provide sentiment specific word counts, the sentiment around ChatGPT is positive in all three lexicons. In the Bing lexicon, out of the 7159 words analyzed, 4021 were positive and 3128 were negative. That’s 893 more positive than negative words. We see the most occurring positive word “free” was tweeted almost 250 times and the most occurring negative word “fake” occurred around 160 times.

In the nrc lexicon, the word counts are higher in the categories with positive sentiments: anticipation, positive, trust, surprise, joy (in descending order).

In the afinn lexicon, the highest word count is in the +1 category, “love”. The even more positive categories, +3 and +4 also have high word counts. The two top +3 words are “excited” and “love” and +4 “amazing” being tweeted 80 times in the ChatGPT dataset.

In bigrams, the mapped connections are how we understand the frequent topics in co-occurring words in the corpus. Instead of analyzing words in isolation, I thought it would be helpful to look at the most common bigrams occurring greater than 10 times in the dataset. We see the result in the final visualization. People seem to be tweeting about the uses of the chatbot more than their feelings about it. I see “artificial intelligence”, “natural language models”, “content creation world”, “email template”, “research academic writing purposes”, “prompt engineers”. Some possible negative sentiment bigrams include “rival bedrock”, “cheat sheet”, “leaving money”, “premium accounts”.

As suspected, there is a range of feelings around this OpenAI technology, but the sentiment is overall positive. It will be interesting to complete another sentiment analysis in several months after more people have had opportunities to engage the powers of ChatGPT.

References

(1) [(PDF) Sentiment Analysis of ChatGPT Using Twitter Data (researchgate.net)](https://www.researchgate.net/publication/370400200_Sentiment_Analysis_of_ChatGPT_Using_Twitter_Data); Adem Korkmaz and Cemal Aktürk/April 2023

(2) [How ChatGPT will revolutionize the economy | MIT Technology Review](https://www.technologyreview.com/2023/03/25/1070275/chatgpt-revolutionize-economy-decide-what-looks-like/) [https://www.technologyreview.com/author/david-rotman/March 25](https://www.technologyreview.com/author/david-rotman/March%2025), 2023

(3) From the NY Times: “Don’t Ban ChatGPT in Schools. Teach With It” by Kevin Roose, January 12, 2023

(4) [#Conversational AI: An Analysis of Public Twitter Sentiment Surrounding ChatGPT (rstudio-pubs-static.s3.amazonaws.com)](https://rstudio-pubs-static.s3.amazonaws.com/1002551_24fdc33e247042fb88b4cb99c2bbff4b.html)/ Jessica Bethea