HULT
INTERNATIONAL
BUSINESS
SCHOOL

Text Analytics and Natural Language Processing Project

Business Insight Report

Submitted to:

Prof. Thomas Kurnicke

Submitted by:

Sejzi Remorque

December 5, 2021

Introduction

The gaming industry is one the most trending industries ever since the invention of video games. Gaming has always been part of humanity's development since the beginning. the advancement of technology, allowed the gaming sector to reach new heights as well as new consoles are invented every now and then. The graphics of these games also improves as years go by. Almost everyone can now play video games, whether it be smartphones, PCs, or consoles as long as they have access to internet, they can download games through online stores. The most popular ones are video games that are played in PCs and consoles. Among the countless video games released each year, video games that emerged to be the greatest of its year are awarded the "Game of the Year" award. Usually, these games could last years and still be played because of their outstanding gameplay and storyline that could leave lasting impression among players.

With that said, these award-winning games, specifically God of war (2018), Sekiro Shadow Die twice(2019), The Last of us II (2020) ,from the last 3 years will be analyzed using text analytics to determine what made these games impressive in today's gamer standards. The data sources will be gathered from twitter and will be approached using the frameworks' sentiment analysis and Ngrams.

Analysis

The God of War(2018) game is the 4th installment in their series God of war (2005). Unlike the previous series that was set in the Greek Mythos, this installment surprised gamers, after a long 8 years of waiting, with its new take on the series. The backstory of the game was about Kratos, a Greek demi-god, on a journey for revenge for the gods. After his revenge on the 3rd installment, the main character journeys into an uncharted mythology which is the Norse Mythos. In this time, a lot of things have happened, and the main character underwent a huge character development and now journeys along with his son. As a father, the game transitions to a more familial father and son adventure. The main character as he journeyed through challenges and new chapter in his life with his son, now passes on to his son all his wisdom from his regrets, pains, life lessons, and hope.

Using the sentiment analysis approach, the top ten general feeling of gamers when playing God of war were memorable, blockbuster, wow, loving, excited, glowing, enjoying, super, stronger, hot which captures the overall emotions. The game's storyline itself is like book or a movie made into a game. With its character developments, gameplay and graphics, it was able to touch on the feelings of nostalgia of every gamer which made it memorable for the fans and the gaming society.

With the bigrams analysis we can see that the most common tweets are "daily reminders" which can mean that they keep on tweeting out God of war as one of the best games of all time. Furthermore, we can also see the words "combat encounters" and "encounter memorable". The god of war has proven to surprise players with exciting encounters of new combat system, monsters, and characters.

On the other hand, Sekiro is a game that is a “souls type” game made by the famous “from software” which pioneered its popularity ever since creating the their iconic dark souls game. This type of game is known to be challenging and it is not recommended for players that loses their patience easily. However, the game does not only boast on its hard gameplay mechanics, but it also features an engaging “dark-themed” storyline. “Souls type game” usually delves with dark themed story line and a challenging gameplay mechanics to assert the idea of “the more you lose the more you get better at the game” as players forces themselves to understand their enemies’ fighting styles as they progress. The rewards for beating each difficult bosses and challenges are an astounding character developments and story progressions. This in turn makes the players strive to reach for the end game.

With the sentiment analysis, we can see in the table below the words: corrupted, die, fall, damage, struggle, kill, hatred, suffer, rage which are all negative emotions. However, looking at the positive sentiments we can see the following: favorite, spirited, saint, love defeat, wonderful etc. We can infer from here that players usually experience multiple loses which then leads to raging. Despite that it is still their favorite game. Meaning they enjoy the challenges and struggle behind the game progression.

With the use of bigrams method, people mostly tweet about the game’s bosses. And on the top of the list, we can see the name of one of the difficult bosses of the games which is called the “corrupted monk”. Aside from this, we can see two words such as “favorite boss” or “48th attempt” which simply gives the idea of a challenging gameplay. If one does not know the gimmick behind this boss’ behavior it can be quite challenging for the players to progress.

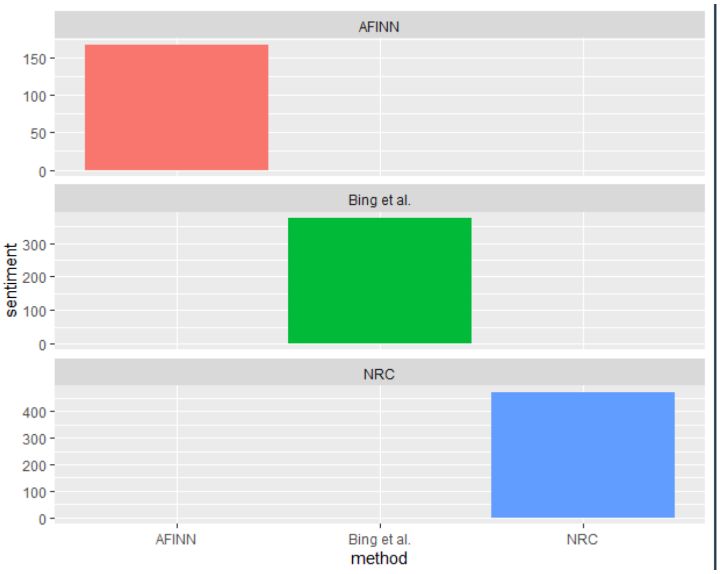
The last of us 2 is a post-apocalyptic zombie game that is the sequel to the last of us. Like any other zombie shows such the walking dead, it delves into the dark side of humans in times of survival and desperation. The game exerts lots of negative emotions as seen in the sentiment analysis. Words such as lost, revenge, lose, cold, pain, traumatic most probably describe the feeling gamers felt throughout the story line. While the word “thoughtful” proves that the deep storyline presented made gamers reflect upon life as seen in the multiple tragic events within the game. The rollercoaster of emotions brought about the sudden plot twists make the gamers feel “spirited” as they dived into the character and story development.

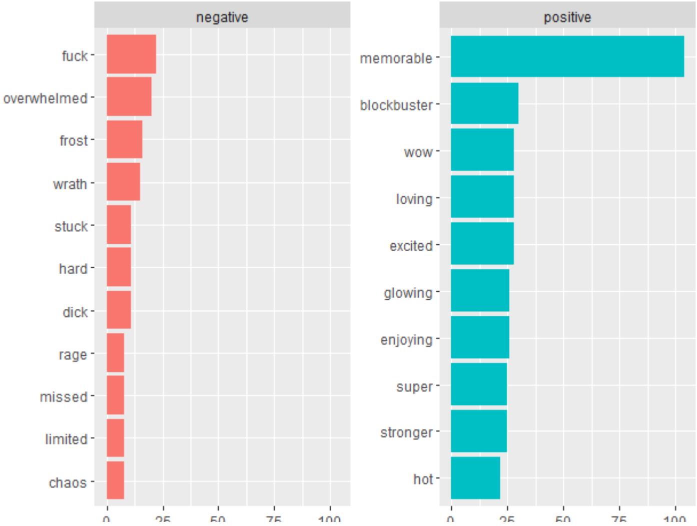
Conclusion

The most obvious reoccurring pattern we can see among these award-winning video games is the “dark themed” storyline. We can infer that most gamers enjoy storylines where the main characters struggle or suffers but leaves the players pondering after reaching the end. We can also see that players also like challenging games where they struggle but feels stronger as they progress throughout. The combat system must be smooth and exciting. Therefore, inputting these elements into a game could actually attract people to play the game.

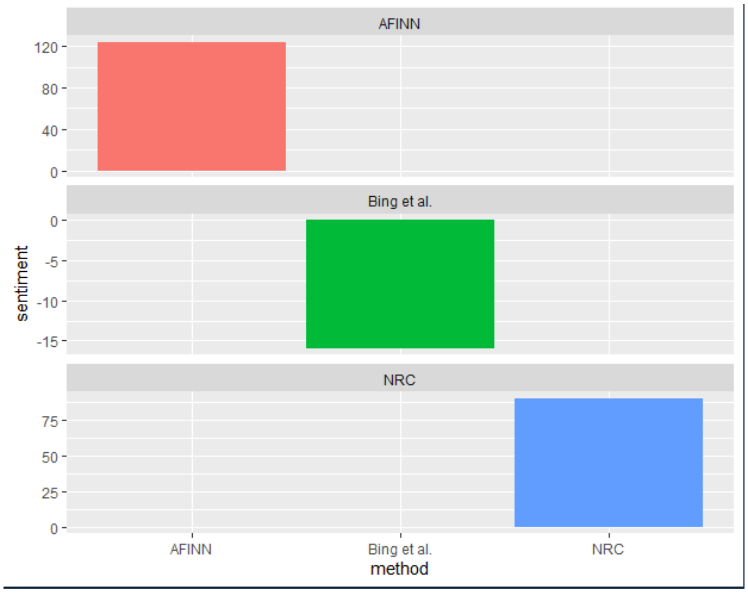
part II. Appendix

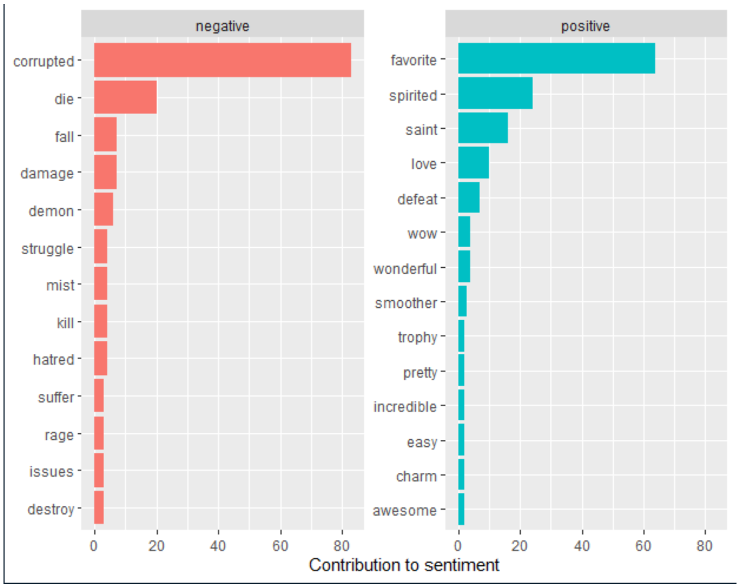
God of War 4 – Sentiment Analysis chart "BING"



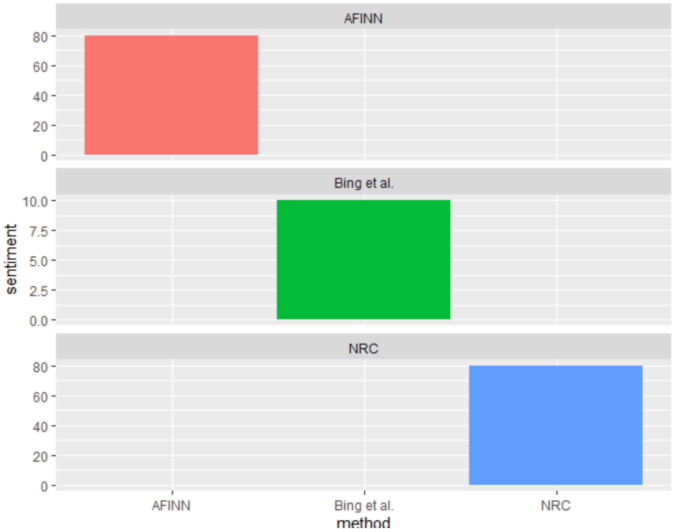


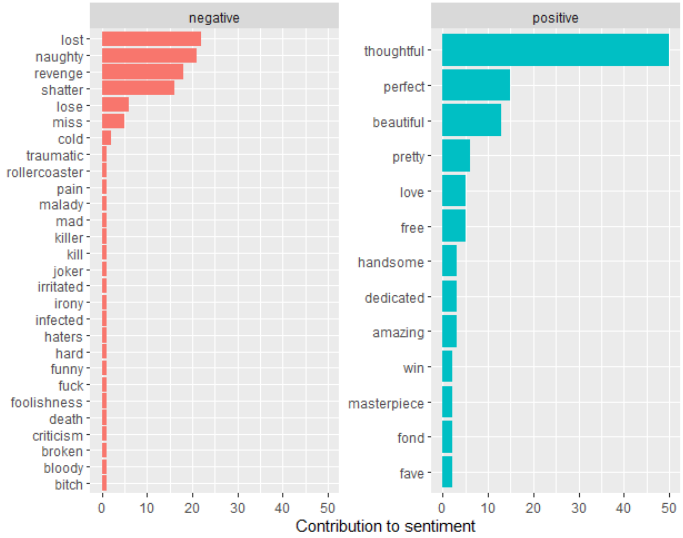
Sekiro – Sentiment Analysis “BING”





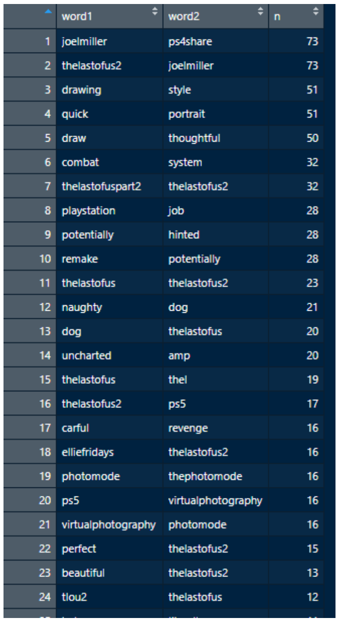
The Last of Us 2 – Sentiment analysis chart





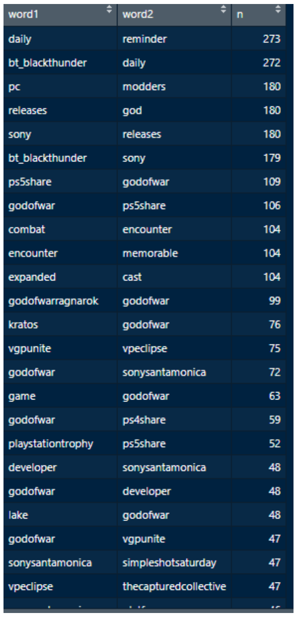
Bigrams

The Last of Us 2:



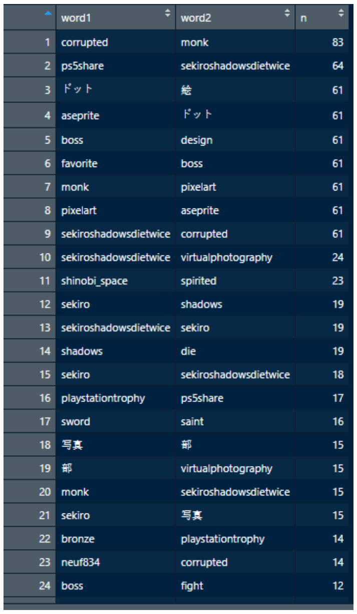
	word1	word2	n
1	joelmiller	ps4share	73
2	thelastofus2	joelmiller	73
3	drawing	style	51
4	quick	portrait	51
5	draw	thoughtful	50
6	combat	system	32
7	thelastofuspart2	thelastofus2	32
8	playstation	job	28
9	potentially	hinted	28
10	remake	potentially	28
11	thelastofus	thelastofus2	23
12	naughty	dog	21
13	dog	thelastofus	20
14	uncharted	amp	20
15	thelastofus	thel	19
16	thelastofus2	ps5	17
17	carful	revenge	16
18	elliefridays	thelastofus2	16
19	photomode	thephotomode	16
20	ps5	virtualphotography	16
21	virtualphotography	photomode	16
22	perfect	thelastofus2	15
23	beautiful	thelastofus2	13
24	tlou2	thelastofus	12

God of War:



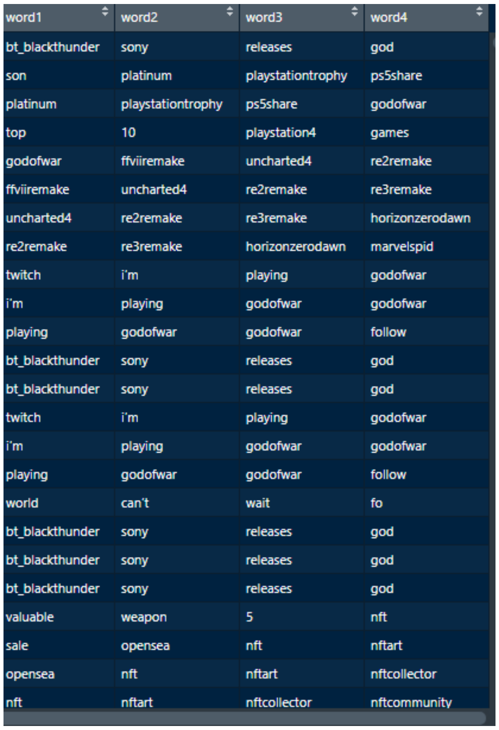
word1	word2	n
daily	reminder	273
bt_blackthunder	daily	272
pc	modders	180
releases	god	180
sony	releases	180
bt_blackthunder	sony	179
ps5share	godofwar	109
godofwar	ps5share	106
combat	encounter	104
encounter	memorable	104
expanded	cast	104
godofwarragnarok	godofwar	99
kratos	godofwar	76
vgpunite	vpeclipse	75
godofwar	sonysantamonica	72
game	godofwar	63
godofwar	ps4share	59
playstationtrophy	ps5share	52
developer	sonysantamonica	48
godofwar	developer	48
lake	godofwar	48
godofwar	vgpunite	47
sonysantamonica	simpleshotsaturday	47
vpeclipse	thecapturedcollective	47

Sekiro:

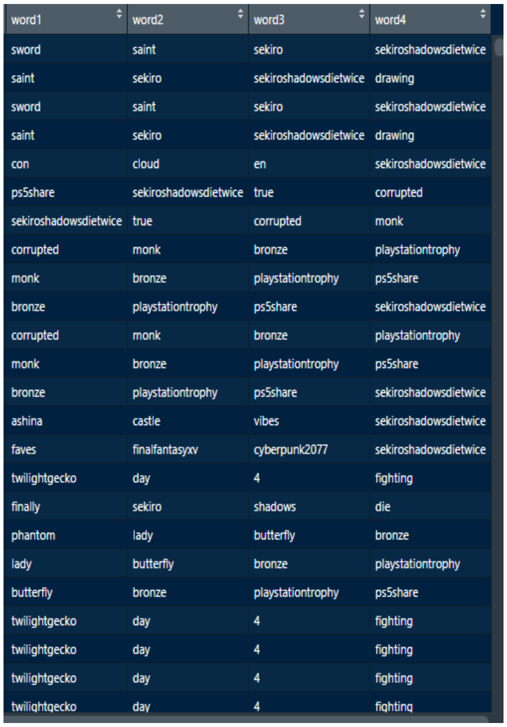


	word1	word2	n
1	corrupted	monk	83
2	ps5share	sekiroshadowsdietwice	64
3	ドット	絵	61
4	aseprite	ドット	61
5	boss	design	61
6	favorite	boss	61
7	monk	pixelart	61
8	pixelart	aseprite	61
9	sekiroshadowsdietwice	corrupted	61
10	sekiroshadowsdietwice	virtualphotography	24
11	shinobi_space	spirited	23
12	sekiro	shadows	19
13	sekiroshadowsdietwice	sekiro	19
14	shadows	die	19
15	sekiro	sekiroshadowsdietwice	18
16	playstationtrophy	ps5share	17
17	sword	saint	16
18	写真	部	15
19	部	virtualphotography	15
20	monk	sekiroshadowsdietwice	15
21	sekiro	写真	15
22	bronze	playstationtrophy	14
23	neuf834	corrupted	14
24	boss	fight	12

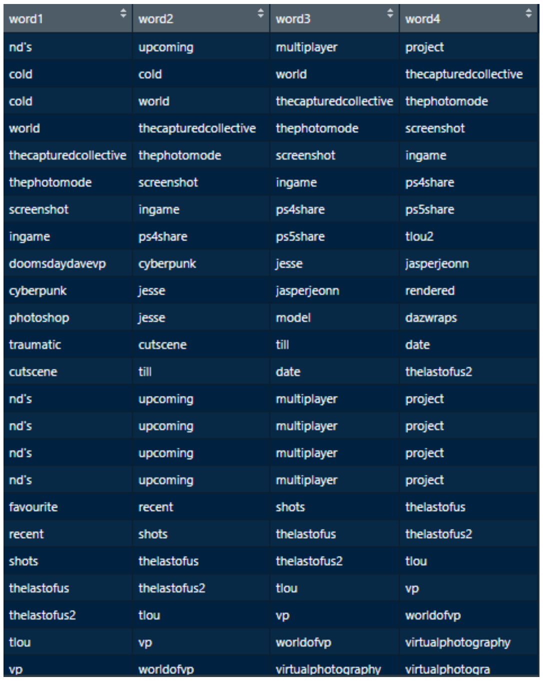
Quadrograms



word1	word2	word3	word4
bt_blackthunder	sony	releases	god
son	platinum	playstationtrophy	ps5share
platinum	playstationtrophy	ps5share	godofwar
top	10	playstation4	games
godofwar	ffviiremake	uncharted4	re2remake
ffviiremake	uncharted4	re2remake	re3remake
uncharted4	re2remake	re3remake	horizonzerodawn
re2remake	re3remake	horizonzerodawn	marvelspid
twitch	i'm	playing	godofwar
i'm	playing	godofwar	godofwar
playing	godofwar	godofwar	follow
bt_blackthunder	sony	releases	god
bt_blackthunder	sony	releases	god
twitch	i'm	playing	godofwar
i'm	playing	godofwar	godofwar
playing	godofwar	godofwar	follow
world	can't	wait	fo
bt_blackthunder	sony	releases	god
bt_blackthunder	sony	releases	god
bt_blackthunder	sony	releases	god
valuable	weapon	5	nft
sale	opensea	nft	nftart
opensea	nft	nftart	nftcollector
nft	nftart	nftcollector	nftcommunity



word1	word2	word3	word4
sword	saint	sekiro	sekiroshadowsdietwice
saint	sekiro	sekiroshadowsdietwice	drawing
sword	saint	sekiro	sekiroshadowsdietwice
saint	sekiro	sekiroshadowsdietwice	drawing
con	cloud	en	sekiroshadowsdietwice
ps5share	sekiroshadowsdietwice	true	corrupted
sekiroshadowsdietwice	true	corrupted	monk
corrupted	monk	bronze	playstationtrophy
monk	bronze	playstationtrophy	ps5share
bronze	playstationtrophy	ps5share	sekiroshadowsdietwice
corrupted	monk	bronze	playstationtrophy
monk	bronze	playstationtrophy	ps5share
bronze	playstationtrophy	ps5share	sekiroshadowsdietwice
ashina	castle	vibes	sekiroshadowsdietwice
faves	finalfantasyxv	cyberpunk2077	sekiroshadowsdietwice
twilightgecko	day	4	fighting
finally	sekiro	shadows	die
phantom	lady	butterfly	bronze
lady	butterfly	bronze	playstationtrophy
butterfly	bronze	playstationtrophy	ps5share
twilightgecko	day	4	fighting
twilightgecko	day	4	fighting
twilightgecko	day	4	fighting
twilightgecko	day	4	fighting



word1	word2	word3	word4
nd's	upcoming	multiplayer	project
cold	cold	world	thecapturedcollective
cold	world	thecapturedcollective	thephotomode
world	thecapturedcollective	thephotomode	screenshot
thecapturedcollective	thephotomode	screenshot	ingame
thephotomode	screenshot	ingame	ps4share
screenshot	ingame	ps4share	ps5share
ingame	ps4share	ps5share	tlou2
doomsdaydavevp	cyberpunk	jesse	jasperjeonn
cyberpunk	jesse	jasperjeonn	rendered
photoshop	jesse	model	dazwraps
traumatic	cutscene	till	date
cutscene	till	date	thelastofus2
nd's	upcoming	multiplayer	project
nd's	upcoming	multiplayer	project
nd's	upcoming	multiplayer	project
nd's	upcoming	multiplayer	project
favourite	recent	shots	thelastofus
recent	shots	thelastofus	thelastofus2
shots	thelastofus	thelastofus2	tlou
thelastofus	thelastofus2	tlou	vp
thelastofus2	tlou	vp	worldofvp
tlou	vp	worldofvp	virtualphotography
vp	worldofvp	virtualphotography	virtualphotogra

R Code Section

```
##################################
#######TEXT ANALYTICS USING TWITTER API
#######SEJZI REMORQUE
#######HULT
##################################



library(quanteda)
library(twitteR)
library(rtweet)
library(dplyr)
library(tm)
library(tidyr)
library(stringr)
library(tidytext)
library(ggplot2)

setwd("C:/Users/sejzi/Desktop/Business insight Report")
#download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")



#connect to twitter API
consumer_key<-"Lpv3MfS4hAe6Y6MC2v85fZa1U"
consumer_secret<-"DKz9UnMaurcv11C4KeXX0WqVRdKhdFonEld0NrcH2f1kQrqTr4"
access_token<-"1407408500179755009-5gqgoA9Q29xQVSjI9M3c454J5R3aCm"
access_secret<-"YoGLbUWCvqyN6FiJjIFZ50GklBrIt1EjeXjma3UyQQClL"
```

```
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)



###Get tweets###

####Game of the year fro the past 4 years



gow4<-twitteR::searchTwitter("GodofWar",n=5000,since="2018-12-01",retryOnRateLimit = 1e3,lang = "en")

a = twitteR::twListToDF(gow4)


sekiro<-twitteR::searchTwitter("SekiroShadowsDieTwice ",n=5000,since="2019-12-01",retryOnRateLimit = 1e3,lang="en")

b = twitteR::twListToDF(sekiro)


tlou2<-twitteR::searchTwitter("TheLastofUs2",n=5000,since="2020-06-01",retryOnRateLimit = 1e3,lang = "en")

c = twitteR::twListToDF(tlou2)



######removing stop words


cust_stop <- data_frame(word=c("http", "https", "rt", "t.io","t.co"),

                      lexicon=rep("cust", each=5)

)
```

```
tidy_gow4 <- a %>%

  unnest_tokens(word, text) %>%

  anti_join(stop_words) %>%

  anti_join(cust_stop)


tidy_sekiro <- b %>%

  unnest_tokens(word, text) %>%

  anti_join(stop_words) %>%

  anti_join(cust_stop)


tidy_tlou2 <- c %>%

  unnest_tokens(word, text) %>%

  anti_join(stop_words) %>%

  anti_join(cust_stop)



#################

###COunting frequencies

########################


gow4_nostop<-tidy_gow4


gow4_nostop %>%

  count(word, sort=TRUE)


freq_hist <-gow4_nostop %>%

  count(word, sort=TRUE) %>%

  filter(n>100) %>% # we need this to eliminate all the low count words

  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_hist)

#####################
####Sentiment Analysis
######################

########################################################
##### Comparing different sentiment ####
########################################################

godofwar <- tidy_gow4

afinn <- godofwar %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

bing_and_nrc <- bind_rows(
  godofwar%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
  godofwar %>%
    inner_join(get_sentiments("nrc") %>%
                  filter(sentiment %in% c("positive", "negative"))) %>%
    mutate(method = "NRC")) %>%
  count(method,  sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive-negative)

bind_rows(afinn, bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")

##Sekiro

sekirod2 <- tidy_sekiro

afinn <- sekirod2 %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

bing_and_nrc <- bind_rows(
  sekirod2%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
  sekirod2%>%
    inner_join(get_sentiments("nrc") %>%
                  filter(sentiment %in% c("positive", "negative"))) %>%
    mutate(method = "NRC")) %>%
  count(method,  sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive-negative)
```

```
bind_rows(afinn, bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")

####Tlou2
lastofus <- tidy_tlou2

afinn <- lastofus %>%
  inner_join(get_sentiments("afinn"))%>%
  summarise(sentiment=sum(value)) %>%
  mutate(method="AFINN")

bing_and_nrc <- bind_rows(
  lastofus%>%
    inner_join(get_sentiments("bing"))%>%
    mutate(method = "Bing et al."),
  lastofus%>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive", "negative"))) %>%
    mutate(method = "NRC")) %>%
  count(method,  sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive-negative)

bind_rows(afinn, bing_and_nrc) %>%
  ggplot(aes(method, sentiment, fill=method))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~method, ncol =1, scales= "free_y")


############################################################
######## Most common positive and negative words #############
############################################################

##God of War
bing_counts_gow4 <- godofwar %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bing_counts_gow4

bing_counts_gow4 %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()



##Sekiro
```

```
bing_counts_sekiro <- sekirod2%>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bing_counts_sekiro

bing_counts_sekiro %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()


###Tlou2

bing_counts_tlou2 <- lastofus%>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bing_counts_tlou2

bing_counts_tlou2 %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word, n)) %>%
  ggplot(aes(word, n, fill=sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y")+
  labs(y="Contribution to sentiment", x=NULL)+
  coord_flip()


#############################################
#########NGRAMS - Bigrams - QUADROGRAMS###########
################################################




######GOD OF WAR

gow4_bigrams <- a %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)

gow4_bigrams #We want to see the bigrams (words that appear together, "pairs")

gow4_bigrams %>%
  count(bigram, sort = TRUE) #this has many stop words, need to remove them

#to remove stop words from the bigram data, we need to use the separate function:
```

```
library(tidyr)

bigrams_separated_gow4 <- gow4_bigrams %>%

  separate(bigram, c("word1", "word2"), sep = " ")


bigrams_filtered_gow4 <- bigrams_separated_gow4 %>%

  filter(!word1 %in% stop_words$word) %>%

  filter(!word2 %in% stop_words$word)%>%

  filter(!word1 %in% cust_stop$word)%>%

  filter(!word2 %in% cust_stop$word)



#creating the new bigram, "no-stop-words":

bigram_counts_gow4 <- bigrams_filtered_gow4  %>%

  count(word1, word2, sort = TRUE)

#want to see the new bigrams

bigram_counts_gow4






#######SEKIRO


sekiro_bigrams <- b %>%

  unnest_tokens(bigram, text, token = "ngrams", n=2)


sekiro_bigrams #We want to see the bigrams (words that appear together, "pairs")


sekiro_bigrams %>%
  count(bigram, sort = TRUE) #this has many stop words, need to remove them

#to remove stop words from the bigram data, we need to use the separate function:
library(tidyr)
bigrams_separated_sekiro <- sekiro_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_filtered_sekiro <- bigrams_separated_sekiro %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)%>%
  filter(!word1 %in% cust_stop$word)%>%
  filter(!word2 %in% cust_stop$word)



#creating the new bigram, "no-stop-words":
bigram_counts_sekiro <- bigrams_filtered_sekiro  %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
bigram_counts_sekiro







#THE LAST OF US 2

tlou2_bigrams <- c %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2)
```

```
tlou2_bigrams #We want to see the bigrams (words that appear together, "pairs")


tlou2_bigrams %>%
  count(bigram, sort = TRUE) #this has many stop words, need to remove them


#to remove stop words from the bigram data, we need to use the separate function:
library(tidyr)
bigrams_separated_tlou2 <- tlou2_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")


bigrams_filtered_tlou2 <- bigrams_separated_tlou2 %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)%>%
  filter(!word1 %in% cust_stop$word)%>%
  filter(!word2 %in% cust_stop$word)



#creating the new bigram, "no-stop-words":
bigram_counts_tlou2 <- bigrams_filtered_tlou2  %>%
  count(word1, word2, sort = TRUE)
#want to see the new bigrams
bigram_counts_tlou2




#######################################
########Quadrograms###############
```

```
################################



###God of War
quadrogram_gow4 <- a %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word) %>%
  filter(!word1 %in% cust_stop$word)%>%
  filter(!word2 %in% cust_stop$word)%>%
  filter(!word3 %in% cust_stop$word)%>%
  filter(!word4 %in% cust_stop$word)

quadrogram_gow4



####Sekiro
quadrogram_sekiro <- b %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word) %>%
  filter(!word1 %in% cust_stop$word)%>%
  filter(!word2 %in% cust_stop$word)%>%
  filter(!word3 %in% cust_stop$word)%>%
  filter(!word4 %in% cust_stop$word)

quadrogram_

###The Last of US

quadrogram <- c %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word) %>%
filter(!word1 %in% cust_stop$word)%>%
filter(!word2 %in% cust_stop$word)%>%
filter(!word3 %in% cust_stop$word)%>%
filter(!word4 %in% cust_stop$word)

quadrogram
```