Business Insight Report

AAA is an informal classification used for video games. Among gamers, AAA masterpiece refers to the high-cost, high-volume, and high-quality single-player games produced by Dachang. Here, we have selected the three top-selling AAA games in the second half of 2020 for text analysis, namely, Cyberpunk 2077, Watchdogs: Legion and Assassin's Creed: Valhalla.

First of all, we chose the twitter R method to get data through api. We chose to obtain only 10,000 data from the second half of the year. From a simple observation, we found that most of these twitter users use Android phones, more use web pages, and a small number use iPhones. Then we start to tokenize the data, we merge the three games into one data frame, and start adding emotional tags, here we choose the bing database. After cleaning up the stop words, we also sorted the words that appeared frequently. We found that the word white has a high frequency, reaching 232. I am not sure if it is related to race. Another point is that the word art also appears 185 Secondly, that is to say, players generally show respect for the art of aaa works, and such games have universal artistic value. In the end, death and cruelty appeared in the top 20, showing that these games are easy to die and the plot is not the comedy that players expect.

Then, we used the tf-idf method to filter out the words with the highest gold content. When sorting, something very interesting appeared. In Watch Dogs: Legion, the word ps5 is very unique. In Assassin’s Creed: Valhalla, the same word points to xbox, and both appear "host". Obviously we can see the popularity of these two games on the console platform. Therefore, in the sales direction, these The series of products should focus more on the host platform. What's interesting is that Cyberpunk 2077 has been discussed in multiple languages, which shows that this product has a wide audience around the world. There are players from all over the world discussing the game and its studio cd project. Similarly, negative words appeared in these words, which also shows that this game has certain shortcomings or problems.

Furthermore, we use word cloud to get the keywords of these three games. These words stand out: anger, tears, negative, surprise, trust, finally, twitch, watch, joy, anticipation. What can tell us is that the game may be very touching, but it does not meet the expectations of the players. The manufacturers have let down the trust of the players. This makes the players very angry because the actual experience is lower than expected. The players have waited for these games for a long time. Many players will Watch the game first on twitch. So in terms of commercial considerations, I think the quality of the works in the second half of the year is far below expectations. Manufacturers should consider more discounts or better patches or dlcs to make up for the lack or lack of game quality.

We drew the term frequency chart. In fact, we can find that the middle parts of the three games are quite consistent in terms of terms, but there are obvious differences in terms of high-rank and low-rank terms, which tells us that we should pay more attention to these High-frequency words, because compared to other data, the high-frequency words here are more representative and analytical. In addition, Assassin’s Creed players care more about the protagonist and art, which means that the gameplay of this work is not that important. What ubisoft needs to do is to pay more attention to the modeling of the scene and the protagonist’s image and character. shape. Cyberpunk developers are more concerned than the game itself,

which shows that the success of this game is inseparable from the reputation and credibility of the developer itself. Watchdog players pay more attention to the ip itself rather than the game, and people will buy the game because of this series, rather than the game itself or the developer.

Regarding bigram, we can see the enthusiasm of players on the two host platforms for game sharing through screening. What's interesting is that players seem to be paying close attention to an artist named donglu yu, who has designed many scenes for Assassin's Creed, including the previous work, Assassin's Creed Black Flag. In these three works, the results of cyberpunk are obviously more frequent. Although the discussion of Assassin's Creed is scattered, it is obviously more extensive, and the total amount is significantly larger than the other two works. Watch Dogs is relatively the least popular.

In the end, I used a quadrogram, but didn't find anything interesting. Therefore, we draw the following conclusions: The aaa works in the second half of 2020 did not perform as expected. Among the three games, Assassin's Creed performed the best, Cyberpunk was second, and Watchdogs was third. Among them, Assassin's Creed and Watchdogs had more host players, but From different hosts, in Assassin's Creed, players care more about the protagonist and emotions and taking pictures, about the artistry of the work. In cyberpunk, players generally pay more attention to the game developer's cd project, and more languages are involved in it, which represents more national player groups. The watchdog Japanese players are more prominent, people are more concerned about the name of the game rather than the content and gameplay of the game. For specific business strategies, we recommend: For the Assassin's Creed series, you can expand more to the ps5 platform or make up for your own xbox sharing deficiencies. For cyberpunk, developers should pay more attention to their own image and credibility. Developers and this game are closely linked and closely related. At best, the watchdog should seriously reflect on this work, it is more like an IP-linked work rather than a successful sequel, and pay more attention to the Japanese market.

Appendix

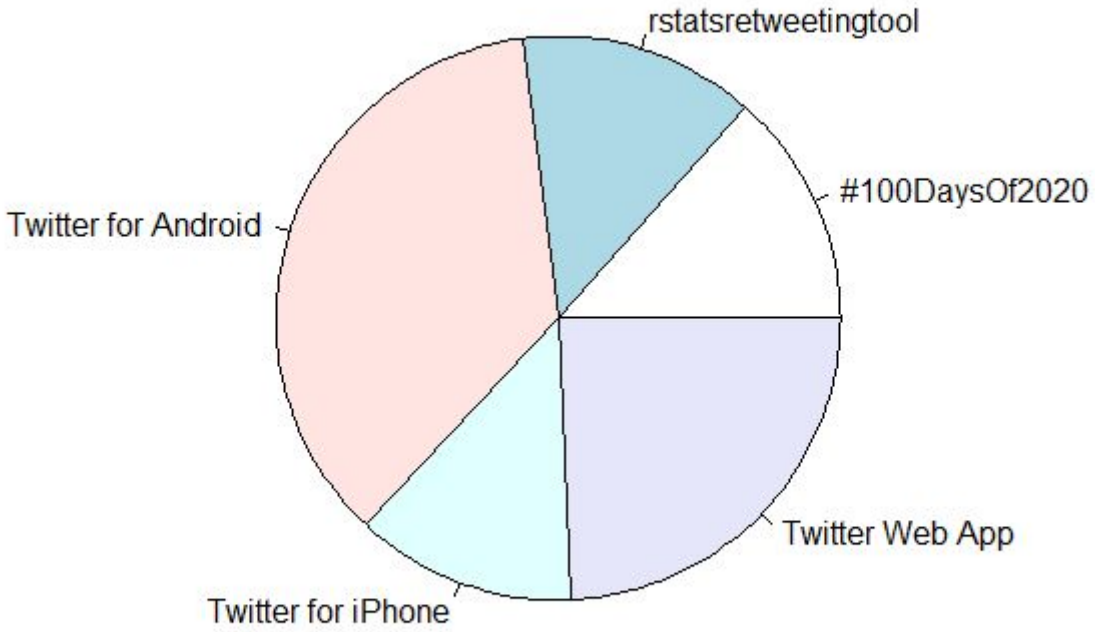
```
##### packages and libraries #####
#install.packages("twitteR")
library(twitteR)
library(tm)
library(textreadr)
library(dplyr)
library(tidytext)
library(tidyverse)
library(magrittr)
library(stringr)
library(ggplot2)
library(wordcloud)

setwd("/Users/RainyVintage/Desktop")
setup_twitter_oauth("JOv7uzKcYWejGPteV5RGa8dJy",
"HgoVIIAjn3bNOZiRT9cd77vjjEwiE0cBKLhhK2qdyCRHfdwyPC","135751944161803469
1-Xo2mj8ClI8voR7AN0ydl4iTbFJ6Ye7","Mm3W4BjemPlkm05VBE5y8KaeMzqcxlOs9Db
bNNrzrMDr0")
#download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

r_tweets <- searchTwitter("#Cyberpunk2077 #Cyberpunk 2077 #WatchDogsLegion
#AssassinsCreedValhalla", n=10000)
sources <- sapply(r_tweets, function(x) x$getStatusSource())
sources <- gsub("</a>", "", sources)
sources <- strsplit(sources, ">")
sources <- sapply(sources, function(x) ifelse(length(x) > 1, x[2], x[1]))
source_table = table(sources)
pie(source_table[source_table > 13])

data("stop_words")

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
Cyberpunk <- twitteR::searchTwitter("#Cyberpunk2077 #Cyberpunk 2077", n = 1000, since
= '2020-01-01', retryOnRateLimit = 1e3)
c = twitteR::twListToDF(Cyberpunk)5

WD_Legion <- twitteR::searchTwitter("#WatchDogsLegion", n = 5000, since = '2020-01-01',
retryOnRateLimit = 1e3)
w = twitteR::twListToDF(WD_Legion)
```

```
AC_Valhalla <- twitteR::searchTwitter("#AssassinsCreedValhalla", n = 5000, since =
'2020-01-01', retryOnRateLimit = 1e3)
a = twitteR::twListToDF(AC_Valhalla)

c$game <- 'Cyberpunk'
w$game <- 'WD_Legion'
a$game <- 'AC_Valhalla'

df = Reduce(function(x, y) merge(x, y, all=TRUE), list(c,w,a))
df = df %>%
  select(text, game)

################################################
### Sentiment wordclounds for Cyberpunk ############
################################################

tidy_game <- df %>%
  unnest_tokens(word, text) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, text, sort=T)

cleaned_game <- tidy_game %>%
  anti_join(stop_words)

cleaned_game %>%
  count(word, sort = TRUE)

bing <- get_sentiments("bing")

### Positive & Negative

gamesentiment <- tidy_game %>%
  inner_join(bing) %>%
  count(game, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)

library(reshape2)
#we need to use the NRC sentiments
tidy_game %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey50"),
```

```
            max.words=50)

#########################################################
###############Tidif framwork#####################
#########################################################

games <- df %>%
  unnest_tokens(word, text) %>%
  count(game, word, sort=TRUE) %>%
  ungroup()

total_words <- games %>%
  group_by(game) %>%
  summarize(total=sum(n))

games <- left_join(games, total_words)

print(games)

ggplot(games, aes(n/total, fill = game))+
        geom_histogram(show.legend=FALSE)+
        xlim(NA, 0.001) +
        facet_wrap(~game, ncol=2, scales="free_y")

#########################################
########## ZIPF's law ################
#########################################

freq_by_rank <- games %>%
  group_by(game) %>%
  mutate(rank = row_number(),
         `term frequency` = n/sum(n))
freq_by_rank

#let's plot ZIPF's Law
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color=game))+
  #let's add a tangent line , the first derivative, and see what the slop is
  geom_abline(intercept=-0.62, slope= -1.1, color='gray50', linetype=2)+
  geom_line(size= 1.1, alpha = 0.8, show.legend = FALSE)+
  scale_x_log10()+
  scale_y_log10()

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

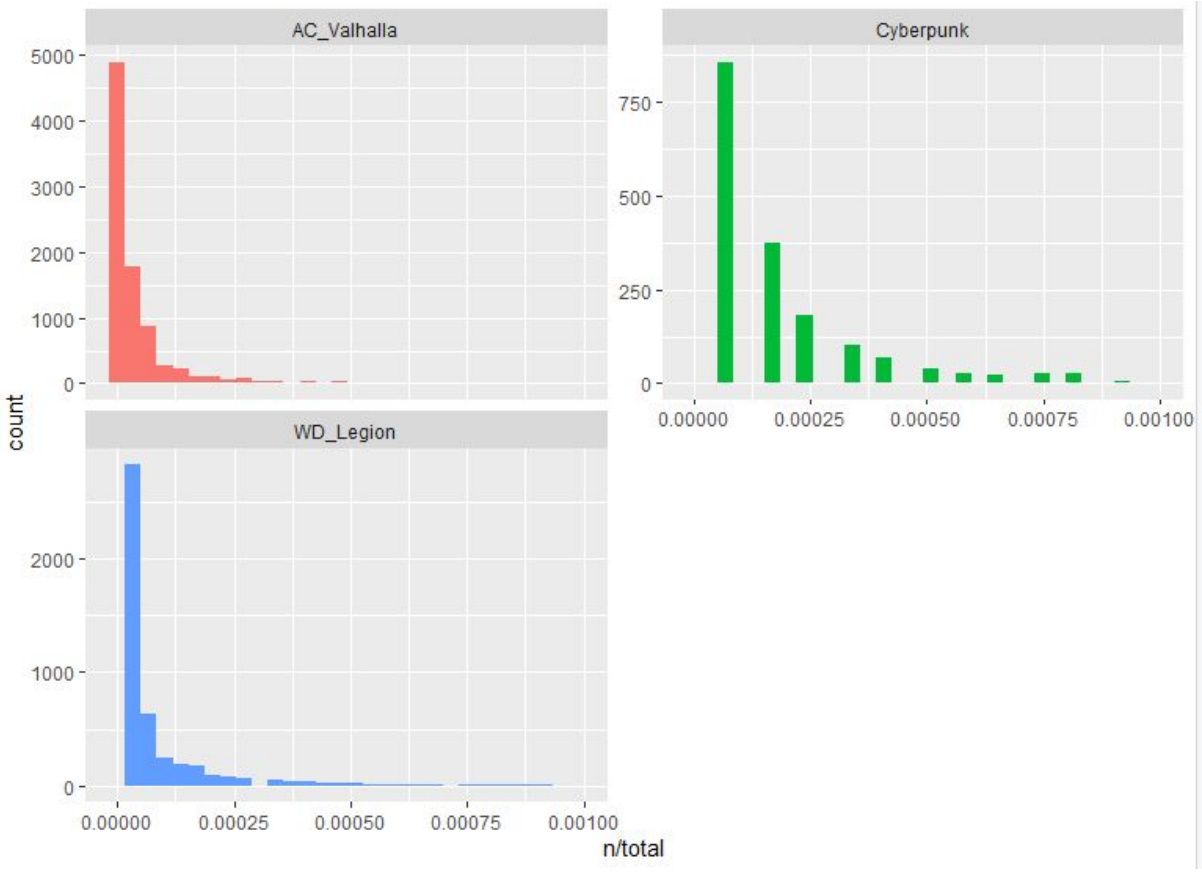
```
################## TF_IDF ##########################
#####################################################

games <- games %>%
  bind_tf_idf(word, game, n)

games

#Graphical apprach:
games %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels =rev(unique(word)))) %>%
  group_by(game) %>%
  top_n(20) %>%
  filter(n<500) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=game))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~game, ncol=2, scales="free")+
  coord_flip()

#############################################################################
############################
########################################################## Bi-grams
#########################################################
#############################################################################
############################

game_bigrams <- df %>%
  unnest_tokens(bigram, text, token ="ngrams", n=2)%>%
  filter(!is.na(bigram))

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

game_bigramgame_bigramss %>%
  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:

bigrams_separated <- game_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_filtered <- bigrams_separated %>%
```

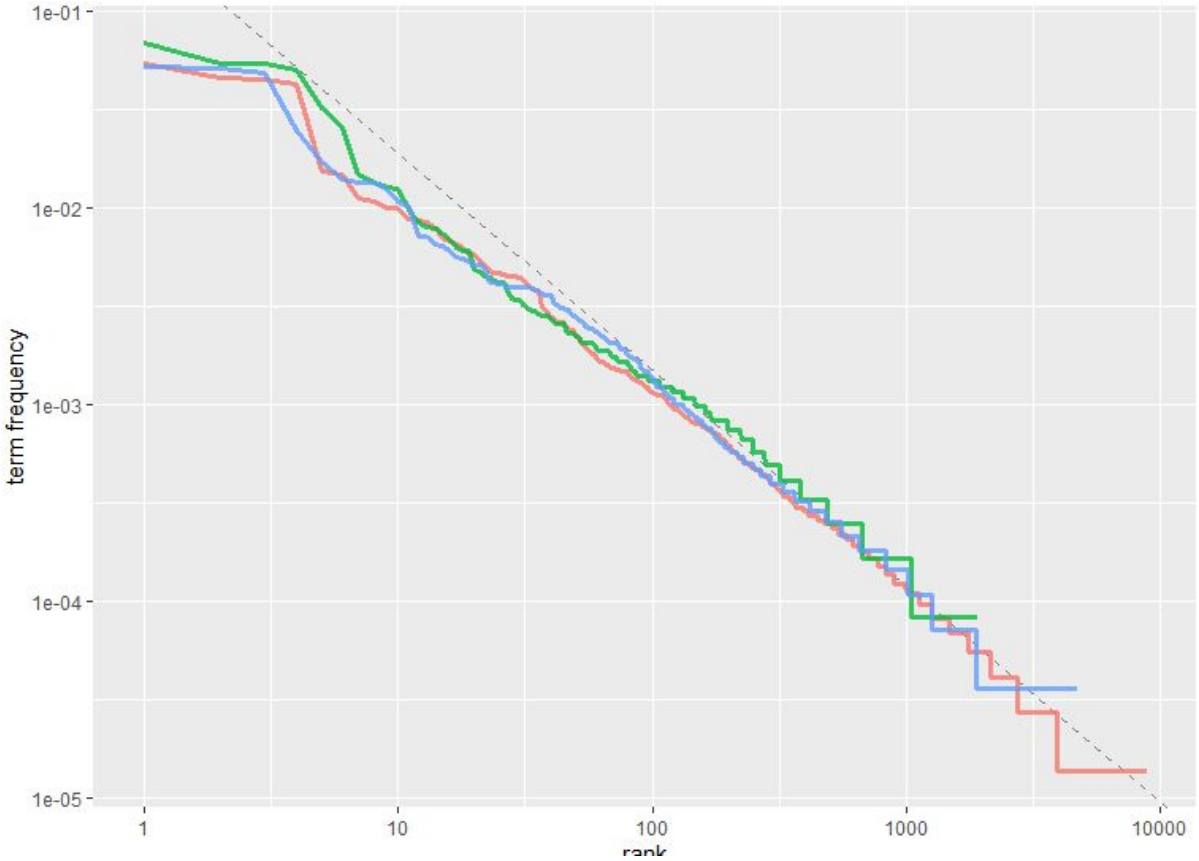
```
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)


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

#####################################################################
###### What if we are interested in the most common #######
################## 4 consecutive words - quadro-gram ########
#####################################################################
quadrogram <- df %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  filter(!is.na(quadrogram)) %>%
  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)

quadrogram
```



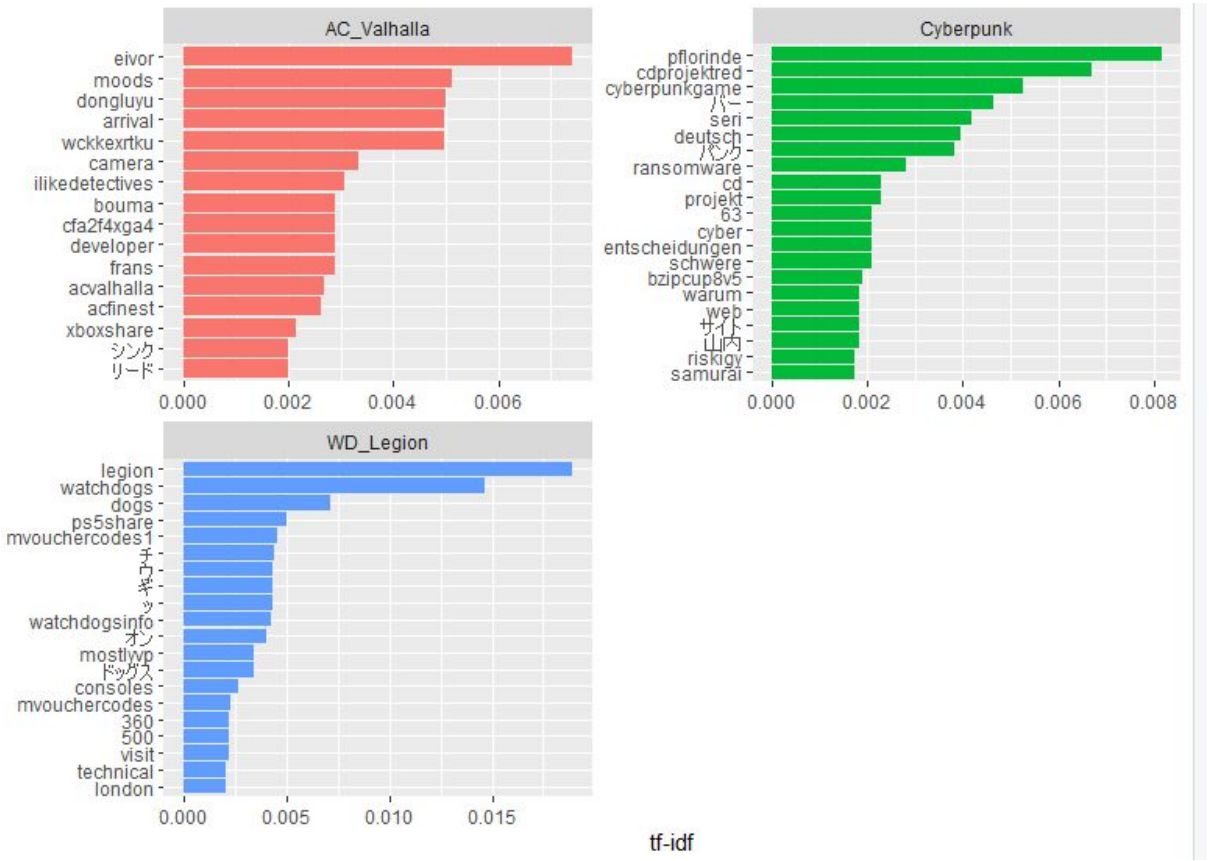

```
+    count(word, sort = TRUE)
          word   n
1        watch 504
2      arrival 330
3      finally 312
4       twitch 243
5        white 232
6         shot 200
7        enjoy 192
8          art 185
9      assassin 160
10        raid 132
11        love 130
12         die 126
13         fun 126
14        deal 125
15        hope 125
16        time 120
17        tree 119
18       cruel 115
19       fight 114
20      pretty 108
```


fear
anticipation
parasite
hoot
feeling
bad
tree
mess
cruel
bug
finally
start
time
found
joy
favorite
beauty
beautiful
love
arrival
anger
pretty
white
watch
attack
hit
enjoy
fun
fight
battle
twitch
series
deal
copy
assassin
raid
negative
visit
chance
trust
full
art
hope
trip
die
good
limited
console
shot
sadness


AC_Valhalla
Cyberpunk
WD_Legion
count
n/total
0
1000
2000
3000
4000
5000
250
500
750
0.00000
0.00025
0.00050
0.00075
0.00100



game	word	n	total	rank	`term frequency`
<chr>	<chr>	<int>	<int>	<int>	<dbl>
1 AC_Valhalla	assassinscreedvalhalla	3941	73051	1	0.0539
2 AC_Valhalla	https	3381	73051	2	0.0463
3 AC_Valhalla	t.co	3288	73051	3	0.0450
4 AC_Valhalla	rt	3139	73051	4	0.0430
5 WD_Legion	https	1451	27849	1	0.0521
6 WD_Legion	t.co	1425	27849	2	0.0512
7 WD_Legion	watchdogslegion	1351	27849	3	0.0485
8 AC_Valhalla	assassinscreed	1127	73051	5	0.0154
9 AC_Valhalla	the	1085	73051	6	0.0149
10 Cyberpunk	cyberpunk	841	12113	1	0.0694




```
> game_bigrams %>%
+   count(bigram, sort = TRUE) #this has many stop words, need to remove them
                                                                bigram    n
1                                                           https t.co 5369
2                                                 assassinscreedvalhalla https  702
3                                                       cyberpunk 2077  585
4                                                        ps4share https  418
5                                                       xboxshare https  386
6                                           assassinscreedvalhalla xboxshare  362
7                                                 for assassinscreedvalhalla  342
8                                                  watchdogslegion ps5share  338
9                                           assassinscreedvalhalla assassinscreed  336
10                                                      rt dongluyu  332
11                                                         done for  331
12                                                    arrival moods  330
13                                                     moods done  330
14                                                   t.co wckkexrtku  330
15                                                  dongluyu arrival  329
```

```
> bigram_counts
                                word1                                word2    n
1                                 https                                t.co 5369
2                   assassinscreedvalhalla                                https  702
3                           cyberpunk                                 2077  585
4                            ps4share                                https  418
5                           xboxshare                                https  386
6                   assassinscreedvalhalla                           xboxshare  362
7                     watchdogslegion                                ps5share  338
8                   assassinscreedvalhalla                           assassinscreed  336
9                                   rt                                dongluyu  332
10                             arrival                                moods  330
11                                t.co                              wckkexrtku  330
12                            dongluyu                                arrival  329
13                            watchdogs                                legion  321
```

```
> quadrogram
          game                 word1                word2                word3                word4
1   AC_Valhalla             assassin's                creed              valhalla          fotovalhalla
2   AC_Valhalla                  creed              valhalla          fotovalhalla              acfinest
3   AC_Valhalla               valhalla          fotovalhalla              acfinest           acphotomode
4   AC_Valhalla          fotovalhalla              acfinest           acphotomode              capturaf8
5   AC_Valhalla              acfinest           acphotomode              capturaf8 assassinscreedvalhalla
6   AC_Valhalla           acphotomode              capturaf8 assassinscreedvalhalla                 https
7   AC_Valhalla              capturaf8 assassinscreedvalhalla                 https                t.co
8   AC_Valhalla assassinscreedvalhalla                 https                t.co           gza9ogbs0o
9   AC_Valhalla                destiny    wolfkissedwednesday          vgpwednesday     virtualphotography
10  AC_Valhalla    wolfkissedwednesday          vgpwednesday     virtualphotography                 https
11  AC_Valhalla          vgpwednesday     virtualphotography                 https                t.co
12  AC_Valhalla     virtualphotography                 https                t.co           sgo1l32y57
13  AC_Valhalla                   xena               scream                it's                monday
14  AC_Valhalla                 scream                 it's              monday                 start
```