Business Insight Report

Phuong-Vy Nguyen

Text Analytics and Natural Language Processing (NLP) - DAT-5317 - FMsBA2

Professor Thomas Kurnicki

Hult International Business School

February 10, 2020

Instagram, Snapchat and Tiktok amongst the top social media platforms and a personal favorite. They are also direct competitors as each try to adopt each other's footsteps whilst maintaining their own distinction. To understand how they set each other apart, a text analysis of the reviews of their mobile applications will be conducted to derive business insights.

To see how similar and/or different Instagram, Snapchat and Tiktok are from one another, Instagram's reviews were used as a benchmark for comparison. A correlation plot was used to visualize the connection between the word frequencies of Instagram-Snapchat and Instagram-Tiktok. The results indicate that Instagram-Snapchat has a higher correlation than Instagram-Tiktok. This is evident in how the Instagram-Snapchat panel is closer to the zero-slope line than in the Instagram-Tiktok panel in the correlogram (see Figure 1). Also, in the Instagram-Snapchat panel, words in the comments section extend to lower frequencies, while there are lightly more empty spaces in the Instagram-Tiktok panel at low frequency. Furthermore, Instagram-Snapchat has a correlation of 0.89, while Instagram-Tiktok has a correlation of 0.77. Thus, these characteristics not only display that Instagram and Snapchat have more similar words in their reviews, but they could potentially offer similar products and are capturing similar customers.

Consequently, it is crucial for Instagram and Snapchat to discover unique keywords that will distinguish the 2 apart and then utilize them to potentially convert existing users from one platform to the other. Tiktok can also adopt a similar approach. The same correlogram has also identified keywords for segmentation. For instance, Instagram reviews have more words like 'action' and 'feed', while Snapchat has 'bitmoji' and 'streak', while Tiktok has 'content' and 'abusive' (see Appendix A).

Moreover, from initial exploration of individual words in all comment sections, words like 'love', 'amazing' and 'nice' were mostly used to describe Instagram and Snapchat. Meanwhile, words such as 'useless', 'worst' and 'waste' were mostly used for TikTok (see Appendix B). If one were to primarily observe these single unit terms, arguably, Instagram and Snacphat have more *positive* reviews than TikTok.

To validate the assumption that Instagram and Snapchat have more *positive* reviews, each word was analyzed with the underlying sentiment as being positive - containing sentiment of joy, or negative - containing sentiment of anger. On one hand, Instagram and Snapchat indeed do have more positive than negative words; as the top term containing homogenous flavors to anger like 'annoying' was used 3,774 times for Instagram and 6,335 times for Snapchat. Meanwhile, the top term containing homogenous flavors to joy like 'love' was mentioned 72,934 times for Instagram and 131,651 times for Snapchat (see Appendix C). On the other hand, Tiktok had the opposite output - with more negative than positive words. For instance, 'bad' - containing flavors of anger - was used the most; 136,880 times to be specific. Meanwhile, 'content' - containing flavors of joy - was only mentioned 44,059 times (see Appendix C).

For concrete business insights, the context of which these single keywords were used must be analyzed with words that tend to follow before or immediately after them, or that tend to co-exist within the same comment section. For example:

Instagram:

When creating relationships between keywords, one can distinctively see phrases like 'enable dark mode/ add dark mode' (see Appendix D). Dark mode is a feature that adjusts the color on one's screen for not only a darker appearance but it also makes the content of the screen stand out more (*Why is Dark Mode so Popular?*, 2020). Instagram was the first social media to implement such a feature, and since then, users seem to be enjoying this ability to highlight information, see images, icons and other elements pop out on their screen.

Users seem to also enjoy the ability to add music to their Instagram story (see Appendix D). Instagram's feature in adding music to the Instagram story feature. Since 2018, Instagram has enabled users to add licensed music to their stories, allowing them "to create a soundtrack to their daily lives" (Viner, 2019). It also integrated Shazam, an application that can identify music and TV shows by listening to a short sample of the audio shortly after. And most recently, it partnered with Spotify to allow seamless song sharing between applications. Given Instagram's consistent efforts to take over the music industry, it is evident Instagram users are appreciating the efforts and in the future, Instagram can utilize this feature to bring benefits to artists who can employ the platform as a marketing strategy.

Snapchat:

Snapchat users 'absolutely love snap'. Snapping is a verb derived from the social media platform name itself; when someone is asked to snap, that means they are being asked to send a photo/video or even message through the app's chatbox feature. Snapchat was made famous for the quick and easy sending/receiving of photos/videos and the crazy filters. The most talked about filter, as evident in the comment section, is the 'gender swap' (see Appendix D). It was amongst the few applications/filters that can successfully and realistically alter one's facial features. Despite the growing interest in turning one's female facial feature to the opposite gender, there was an uproar as to how the filter was exposing the internet's casual transphobia (Anderson, 2019). Thus, to satisfy all users, Snapchat has to figure out how to walk on the fine line between light-hearted fun and creativity, and inclusivity.

Interestingly, there were some interesting phrases that stood out such as 'bad camera quality' and 'constantly app crash' (see Appendix D). The issue of low camera and app crash has been on-going since 2017 (Baral, 2017). This issue is specifically crucial for Android users; as evident in common phrases like 'note 5 app crashes' and 'google pixel 3' - referring to the exact Android models incompatible with the application. Clearly, this is an area that sets Snapchat apart from Instagram and Tiktok but perhaps, turning this into actionable items can yield more positive distinctions.

Tiktok:

Most interestingly, when providing context to unique keywords for Tiktok, one can see phrases like 'sabse ghatiya app' - which is Hindi for 'from all, poor app' (see Appendix D). In fact, a lot of phrases were in Hindi except for a few exceptions like 'google play store' (see Appendix D). This is because most recently, Tiktok underwent a huge scandal that caused their rating on Google Play Store to drop from 4.4 to 1.2. In March 2020, an Indian creator posted a video of himself threatening a woman who had left him; he threw water at her face and the next scene showed her makeup melting and that was mistook as bruising from an acid attack (*Google* *deletes millions of negative TikTok reviews*, 2020). This led to chaos in India, with people commenting to boycott the platform for its lack of protection towards viewers and violent content creation. Google intervened by removing negative reviews to bring back Tiktok's 4.4 stars rating after it determined that there were fake accounts set up to amplify the protest. Despite Google's efforts, Tiktok's ratings still remain below 2 stars on all Android marketplaces (*Google deletes over 80 lakh negative reviews to bring TikTok's rating to 4.4 stars*, 2020). From this insight, it is clear that Tiktok needs major rebranding, specifically in India.

Lastly, it is important to provide a disclaimer that comments in Hindi were written in the English alphabet form, not in scripts, or was written as Hinglish (a combination of Hindi and English). Therefore, it was difficult to clean using R as there was no available package that allowed the translation of Hindi in English alphabet or Hinglish.

In conclusion, analyzing the reviews of the 3 social media platforms - Instagram, Snapchat and Tiktok - provided great insights as to what can be done using specific keywords. Unique words can be utilized as keyword segmentation to connect the platform with their users. And when grouping those unique keywords - seeing them in relations with other words, the context provided from grouped keywords will allow users - see how each platform is distinctive from another, and allow platforms to convert common phrases/ feedback into actionable items.

REFERENCES

Baral, S. (2017, June 9). Snapchat Keeps Crashing: How to fix and troubleshoot the app when it's not working. Mic. Retrieved February 11, 2021, from https://www.mic.com/articles/179443/snapchat-keeps-crashing-how-to-fix-and-troubleshoot-the-app-when-its-not-working

Google deletes millions of negative TikTok reviews. (2020, May 26). BBC News. https://www.bbc.com/news/technology-52808177

Google deletes over 80 lakh negative reviews to bring TikTok's rating to 4.4 stars. The Times of India. (2020, May 29). Retrieved February 11, 2021, from https://timesofindia.indiatimes.com/gadgets-news/google-deletes-over-80-lakh-negative-reviews-to-bring-tiktoks-rating-to-4-4-stars/articleshow/76075093.cms

Anderson, S. (2019, May 16). Snapchat's "gender-swap" filter exposes the internet's casual transphobia. (n.d.). Retrieved February 11, 2021, from https://mashable.com/article/snapchat-gender-swap-filter/

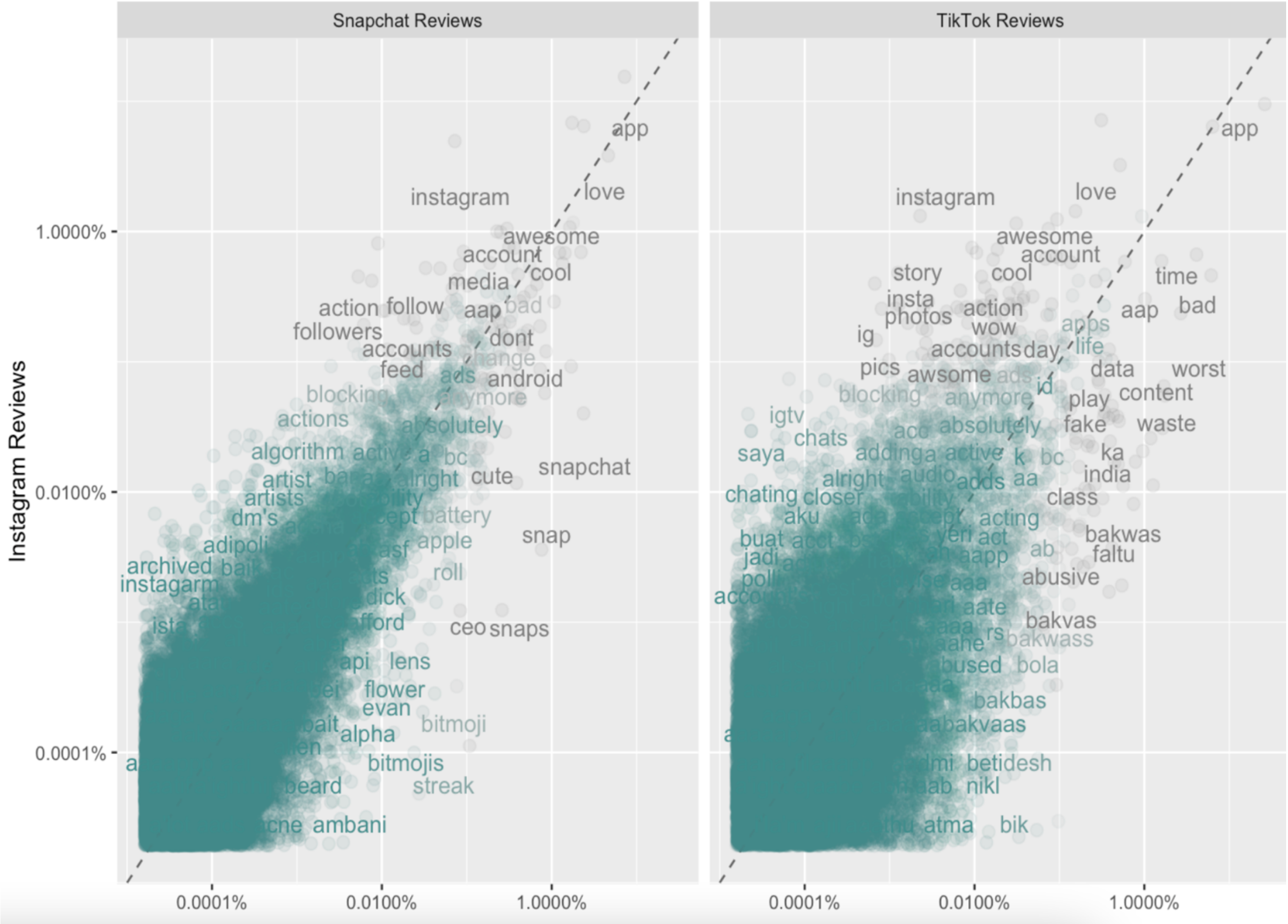
Viner, J. (2019, December 21). Why Instagram Is Taking Over The Music Industry. [And How Artists Are Utilizing The Platform In Their Marketing Strategy.] | by Josh Viner | The Dopamine Effect | Medium. Retrieved February 11, 2021, from https://medium.com/the-dopamine-effect/why-instagram-is-taking-over-the-music-industry-bf4453e35ce3

Why is Dark Mode so Popular?. Wiredelta. (2020, April 14). Retrieved February 11, 2021, from https://wiredelta.com/why-is-dark-mode-so-popular/

APPENDICES

Appendix A

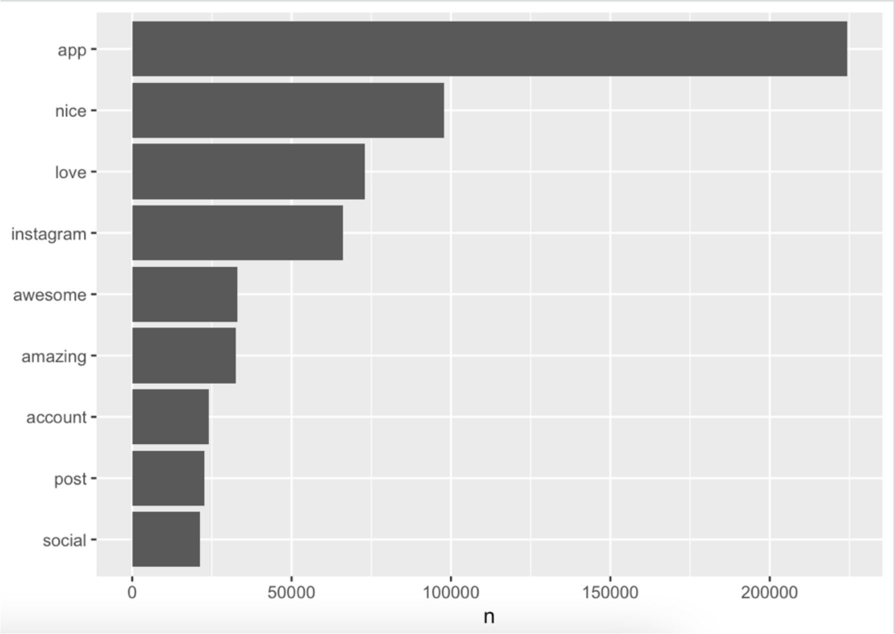
Correlogram between Instagram-Snapchat and Instagram-Tiktok



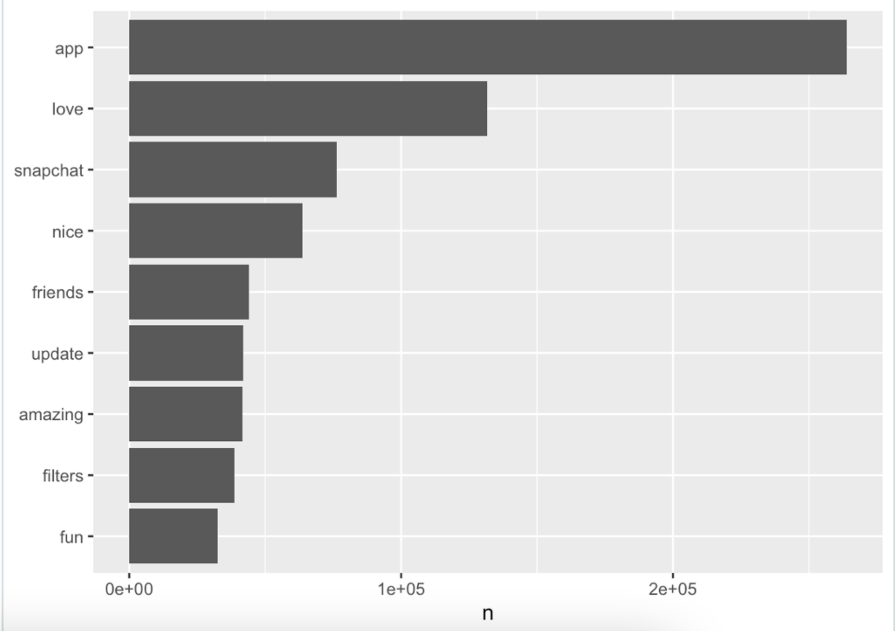
Appendix B

Histograms of Top Frequently Used Words in Comments for Instagram, Snapchat and Tiktok

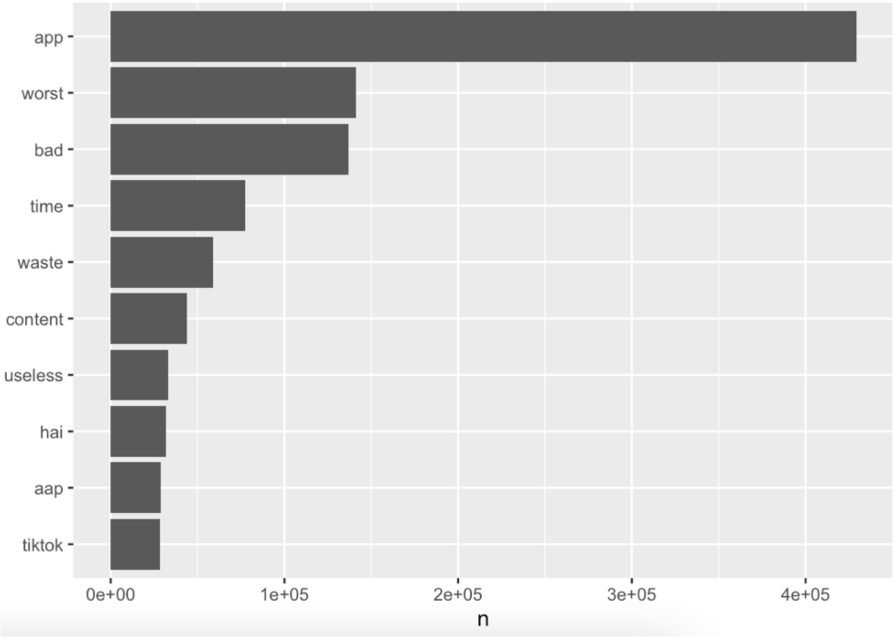
Instagram:



Snapchat:



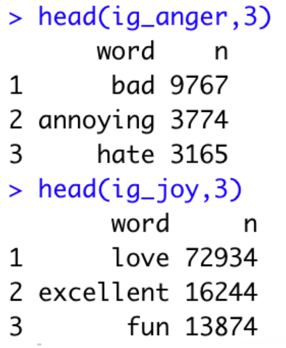
Tiktok:


app
worst
bad
time
waste
content
useless
hai
aap
tiktok
0e+00
1e+05
2e+05
3e+05
4e+05
n

Appendix C

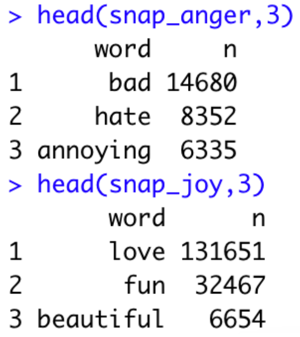
List of Top Words Associated with Anger and Joy Sentiments for Instagram, Snapchat and Tiktok

Instagram:



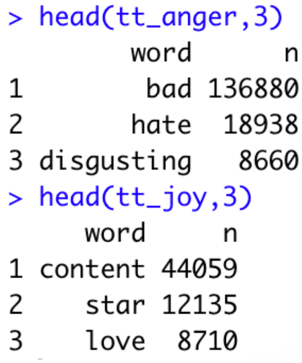
```
> head(ig_anger,3)
      word    n
1      bad 9767
2 annoying 3774
3     hate 3165
> head(ig_joy,3)
       word     n
1      love 72934
2 excellent 16244
3       fun 13874
```

Snapchat:



```
> head(snap_anger,3)
      word     n
1      bad 14680
2     hate  8352
3 annoying  6335
> head(snap_joy,3)
       word      n
1      love 131651
2       fun  32467
3 beautiful   6654
```

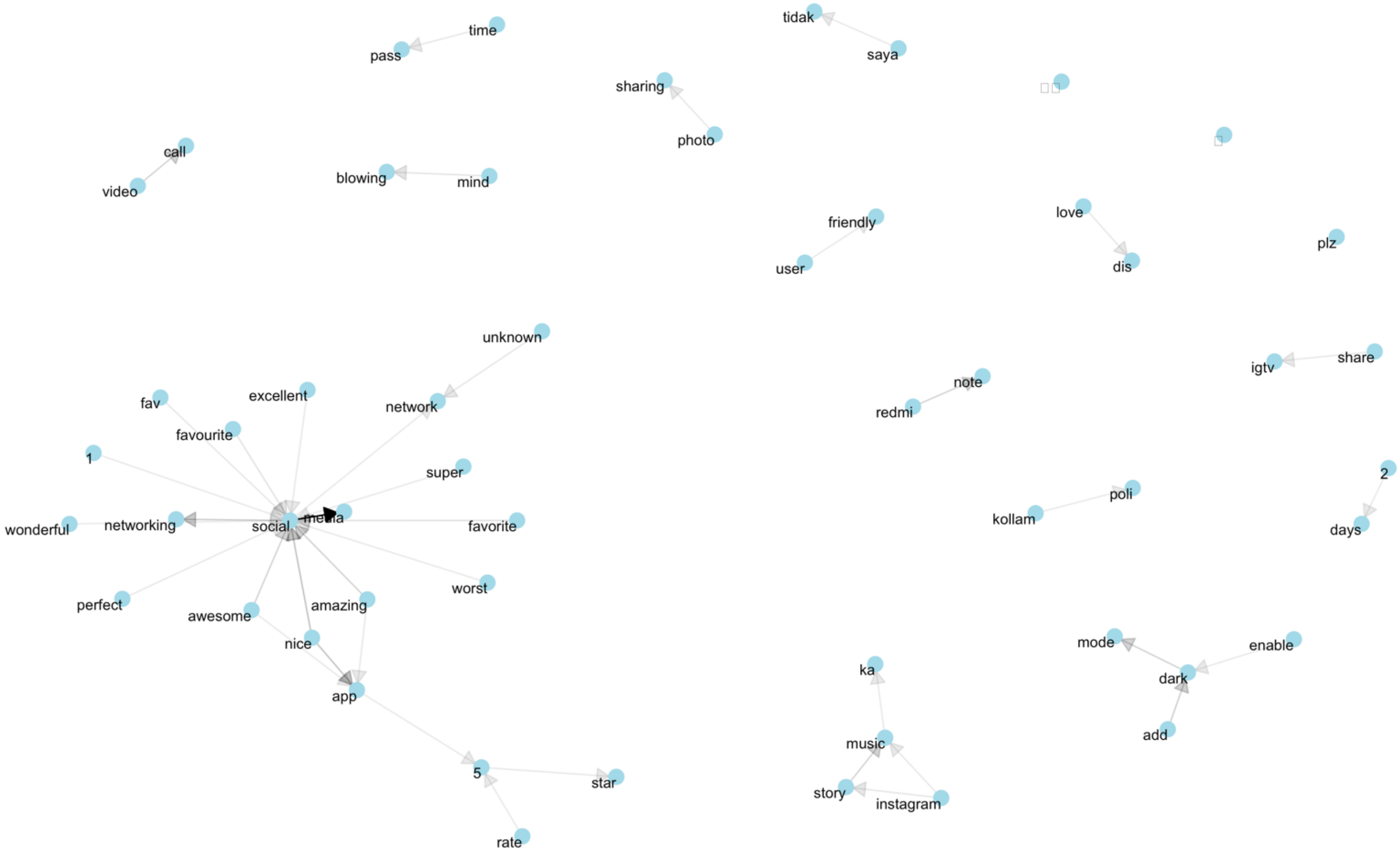
Tiktok:



```
> head(tt_anger,3)
        word      n
1        bad 136880
2       hate  18938
3 disgusting   8660
> head(tt_joy,3)
     word     n
1 content 44059
2    star 12135
3    love  8710
```

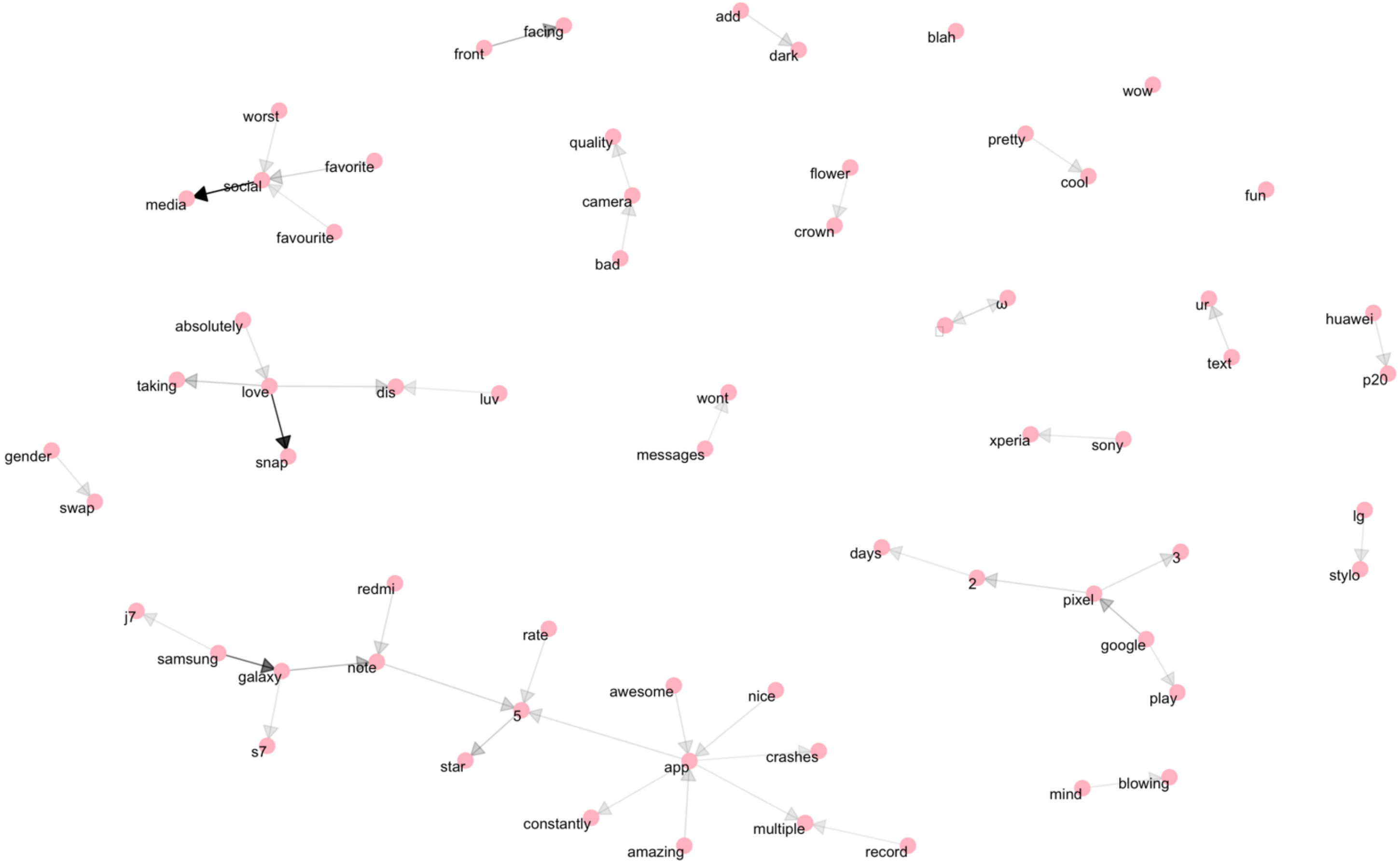
Appendix D

Trigram of Words in Instagram Comment Section



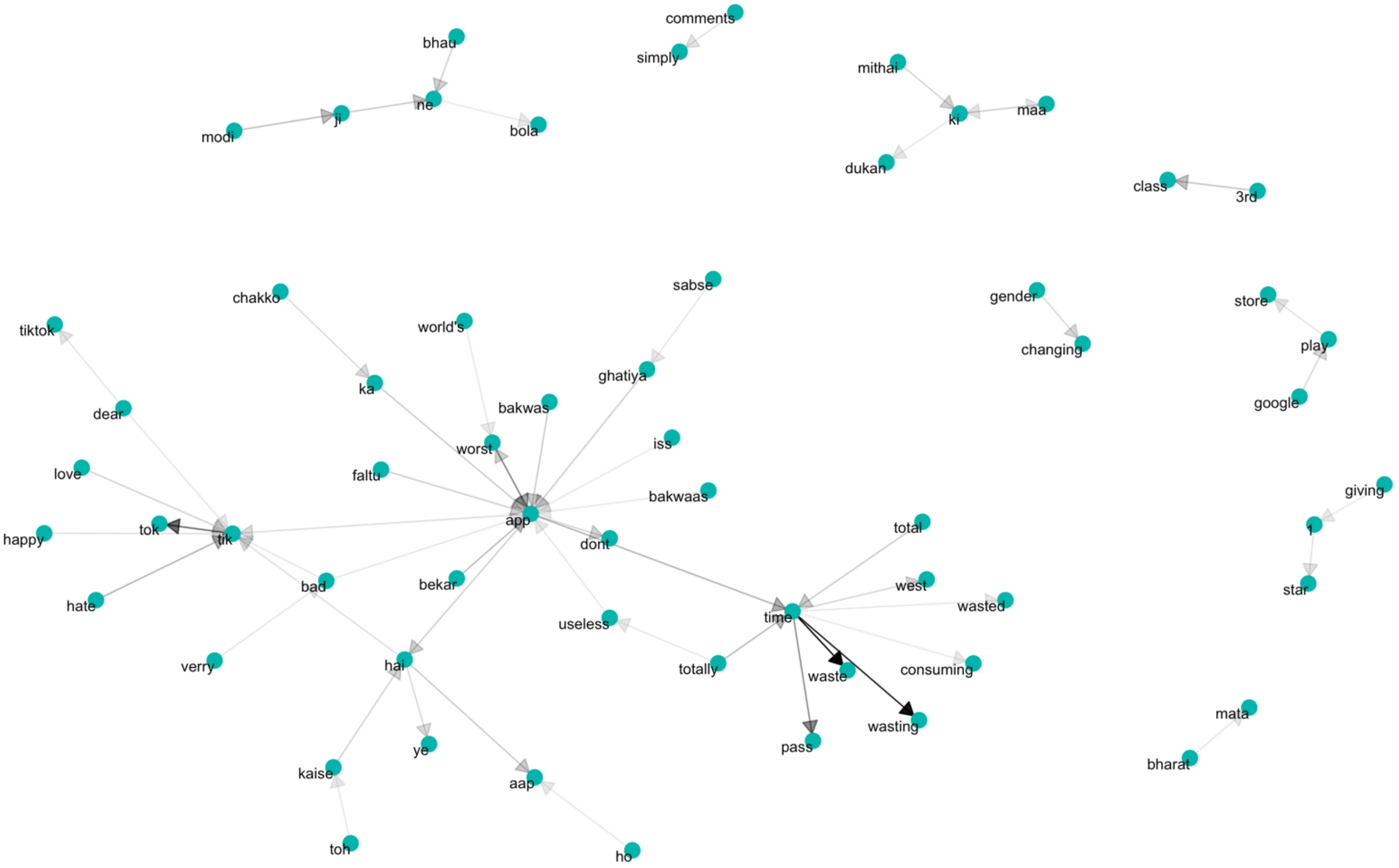
Appendix E

Trigram of Words in Snapchat Comment Section



Appendix F

Trigram of Words in Tiktok Comment Section

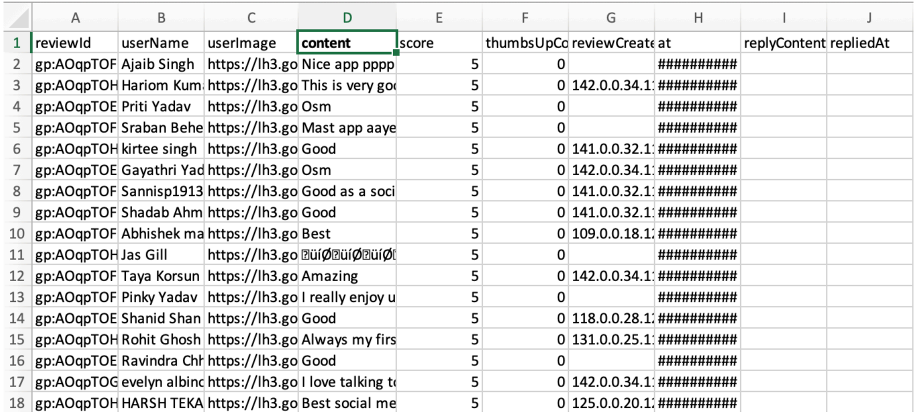


Appendix G

1 million of mobile application reviews for Instagram, Snapchat and Tiktok

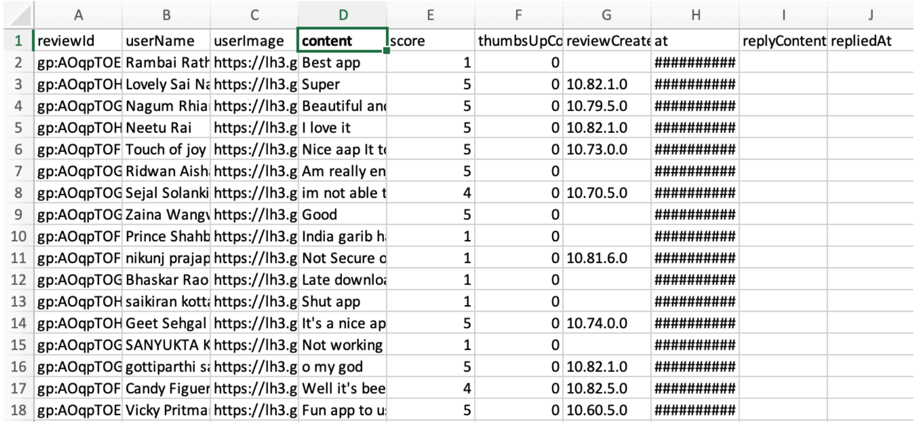
Dataset retrieved from: https://www.kaggle.com/magnumashish/social-media-app-reviews-with-rating

Instagram:



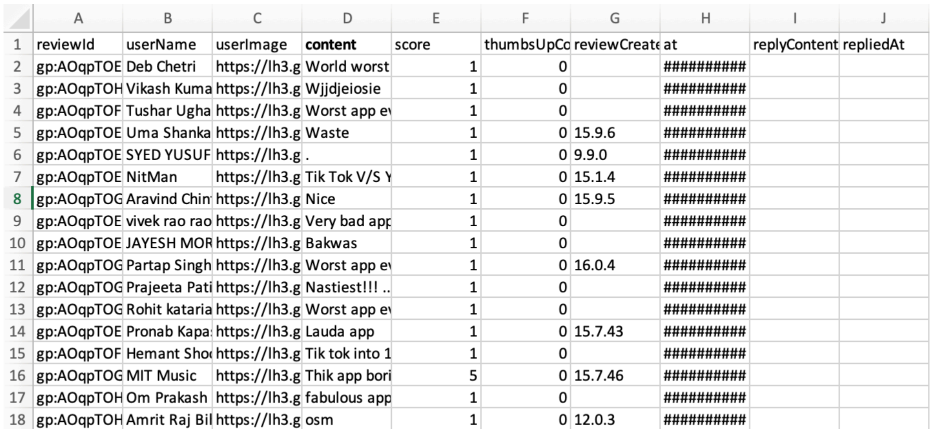
	A	B	C	D	E	F	G	H	I	J
1	reviewId	userName	userImage	**content**	score	thumbsUpCo	reviewCreate	at	replyContent	repliedAt
2	gp:AOqpTOF	Ajaib Singh	https://lh3.go	Nice app pppp	5	0		##########		
3	gp:AOqpTOH	Hariom Kum	https://lh3.go	This is very go	5	0	142.0.0.34.1	##########		
4	gp:AOqpTOE	Priti Yadav	https://lh3.go	Osm	5	0		##########		
5	gp:AOqpTOF	Sraban Behe	https://lh3.go	Mast app aaye	5	0		##########		
6	gp:AOqpTOH	kirtee singh	https://lh3.go	Good	5	0	141.0.0.32.1	##########		
7	gp:AOqpTOE	Gayathri Yad	https://lh3.go	Osm	5	0	142.0.0.34.1	##########		
8	gp:AOqpTOF	Sannisp1913	https://lh3.go	Good as a soci	5	0	141.0.0.32.1	##########		
9	gp:AOqpTOF	Shadab Ahm	https://lh3.go	Good	5	0	141.0.0.32.1	##########		
10	gp:AOqpTOF	Abhishek ma	https://lh3.go	Best	5	0	109.0.0.18.1	##########		
11	gp:AOqpTOH	Jas Gill	https://lh3.go	⍰üíØ⍰üíØ⍰üíØ	5	0		##########		
12	gp:AOqpTOF	Taya Korsun	https://lh3.go	Amazing	5	0	142.0.0.34.1	##########		
13	gp:AOqpTOF	Pinky Yadav	https://lh3.go	I really enjoy u	5	0		##########		
14	gp:AOqpTOE	Shanid Shan	https://lh3.go	Good	5	0	118.0.0.28.1	##########		
15	gp:AOqpTOH	Rohit Ghosh	https://lh3.go	Always my firs	5	0	131.0.0.25.1	##########		
16	gp:AOqpTOE	Ravindra Chh	https://lh3.go	Good	5	0		##########		
17	gp:AOqpTOG	evelyn albinc	https://lh3.go	I love talking t	5	0	142.0.0.34.1	##########		
18	gp:AOqpTOH	HARSH TEKA	https://lh3.go	Best social me	5	0	125.0.0.20.1	##########		

Snapchat:



	A	B	C	D	E	F	G	H	I	J
1	reviewId	userName	userImage	**content**	score	thumbsUpCo	reviewCreate	at	replyContent	repliedAt
2	gp:AOqpTOE	Rambai Rath	https://lh3.g	Best app	1	0		##########		
3	gp:AOqpTOH	Lovely Sai Na	https://lh3.g	Super	5	0	10.82.1.0	##########		
4	gp:AOqpTOG	Nagum Rhia	https://lh3.g	Beautiful and	5	0	10.79.5.0	##########		
5	gp:AOqpTOH	Neetu Rai	https://lh3.g	I love it	5	0	10.82.1.0	##########		
6	gp:AOqpTOF	Touch of joy	https://lh3.g	Nice aap It t	5	0	10.73.0.0	##########		
7	gp:AOqpTOG	Ridwan Aish	https://lh3.g	Am really en	5	0		##########		
8	gp:AOqpTOG	Sejal Solanki	https://lh3.g	im not able t	4	0	10.70.5.0	##########		
9	gp:AOqpTOG	Zaina Wang	https://lh3.g	Good	5	0		##########		
10	gp:AOqpTOF	Prince Shahb	https://lh3.g	India garib h	1	0		##########		
11	gp:AOqpTOF	nikunj prajap	https://lh3.g	Not Secure o	1	0	10.81.6.0	##########		
12	gp:AOqpTOG	Bhaskar Rao	https://lh3.g	Late downlo	1	0		##########		
13	gp:AOqpTOH	saikiran kott	https://lh3.g	Shut app	1	0		##########		
14	gp:AOqpTOH	Geet Sehgal	https://lh3.g	It's a nice ap	5	0	10.74.0.0	##########		
15	gp:AOqpTOG	SANYUKTA K	https://lh3.g	Not working	1	0		##########		
16	gp:AOqpTOG	gottiparthi s	https://lh3.g	o my god	5	0	10.82.1.0	##########		
17	gp:AOqpTOF	Candy Figuer	https://lh3.g	Well it's bee	4	0	10.82.5.0	##########		
18	gp:AOqpTOE	Vicky Pritma	https://lh3.g	Fun app to u	5	0	10.60.5.0	##########		

Tiktok:



	A	B	C	D	E	F	G	H	I	J
1	reviewId	userName	userImage	**content**	score	thumbsUpCo	reviewCreate	at	replyContent	repliedAt
2	gp:AOqpTOE	Deb Chetri	https://lh3.g	World worst	1	0		##########		
3	gp:AOqpTOH	Vikash Kuma	https://lh3.g	Wjjdjeiosie	1	0		##########		
4	gp:AOqpTOF	Tushar Ugha	https://lh3.g	Worst app e	1	0		##########		
5	gp:AOqpTOE	Uma Shanka	https://lh3.g	Waste	1	0	15.9.6	##########		
6	gp:AOqpTOE	SYED YUSUF	https://lh3.g	.	1	0	9.9.0	##########		
7	gp:AOqpTOE	NitMan	https://lh3.g	Tik Tok V/S Y	1	0	15.1.4	##########		
8	gp:AOqpTOG	Aravind Chin	https://lh3.g	Nice	1	0	15.9.5	##########		
9	gp:AOqpTOE	vivek rao rao	https://lh3.g	Very bad app	1	0		##########		
10	gp:AOqpTOE	JAYESH MOR	https://lh3.g	Bakwas	1	0		##########		
11	gp:AOqpTOG	Partap Singh	https://lh3.g	Worst app e	1	0	16.0.4	##########		
12	gp:AOqpTOG	Prajeeta Pati	https://lh3.g	Nastiest!!! ..	1	0		##########		
13	gp:AOqpTOG	Rohit kataria	https://lh3.g	Worst app e	1	0		##########		
14	gp:AOqpTOE	Pronab Kapa	https://lh3.g	Lauda app	1	0	15.7.43	##########		
15	gp:AOqpTOF	Hemant Sho	https://lh3.g	Tik tok into 1	1	0		##########		
16	gp:AOqpTOG	MIT Music	https://lh3.g	Thik app bori	5	0	15.7.46	##########		
17	gp:AOqpTOH	Om Prakash	https://lh3.g	fabulous app	1	0		##########		
18	gp:AOqpTOH	Amrit Raj Bil	https://lh3.g	osm	1	0	12.0.3	##########		

Appendix H
R Code and R Code Output

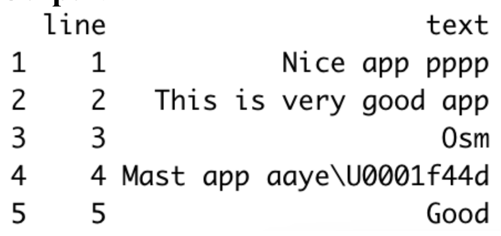
1. IMPORT PACKAGES AND LOAD LIBRARIES

```
#loading all libraries
library(dplyr)
library(tidytext)
library(tidyverse)
library(stringr)
library(tidyr)
library(scales)
library(ggplot2)

#specifying all .cvs files from one directory
setwd("/Users/vynguyen/Downloads/Text Analytics/social media app reviews")

#loading stop_words to clean data later
data(stop_words)
```

2. PREPARE DATASET

```
#filtering .csv dataset for text column only
#turning text column into df

#INSTAGRAM
review = read.csv("Instagram_1Million_reviews.csv", stringsAsFactors = FALSE)
content <- review$content
ig_df <- data.frame(line=1:1000000, text=content)
head(ig_df,5) #load results
```

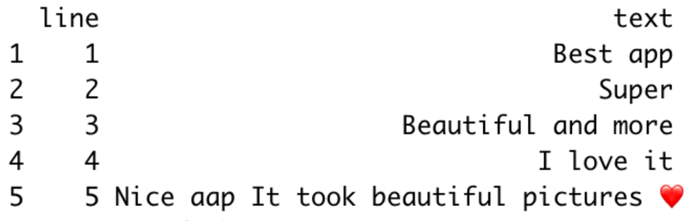
Output:



```
  line                    text
1    1           Nice app pppp
2    2   This is very good app
3    3                     Osm
4    4 Mast app aaye\U0001f44d
5    5                    Good
```

```
#SNAPCHAT
review = read.csv("Snapchat_1Million_reviews.csv", stringsAsFactors = FALSE)
content <- review$content
snap_df <- data.frame(line=1:1000000, text=content)
head(snap_df,5) #load results
```

Output:

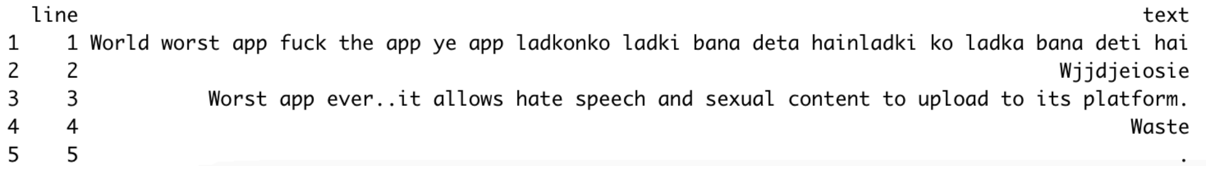


```
  line                                    text
1    1                                Best app
2    2                                   Super
3    3                      Beautiful and more
4    4                               I love it
5    5 Nice aap It took beautiful pictures ❤️
```

#TIKTOK

review = read.csv("TicToc_1Million_reviews.csv", stringsAsFactors = FALSE)
content <- review$content
tt_df <- data.frame(line=1:1000000, text=content)
head(snap_df, 5) *#load results*

Output:



```
  line                                                                        text
1    1 World worst app fuck the app ye app ladkonko ladki bana deta hainladki ko ladka bana deti hai
2    2                                                                     Wjjdjeiosie
3    3          Worst app ever..it allows hate speech and sexual content to upload to its platform.
4    4                                                                           Waste
5    5                                                                               .
```

3. TOKENIZING & COUNTING TOKEN FREQUENCIES FOR ALL 3 DFs

#INSTAGRAM

#tokenizing one token per row, removing stop-words

```
tidy_ig <- ig_df %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words)
```

#counting frequencies for tokens

```
tidy_ig %>%
  count(word, sort=TRUE)
```

#SNAPCHAT

#tokenizing one token per row, removing stop-words

```
tidy_snap <- snap_df %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words)
```

#counting frequencies for tokens

```
tidy_snap %>%
  count(word, sort=TRUE)
```

#TIKTOK

#tokenizing one token per row, removing stop-words

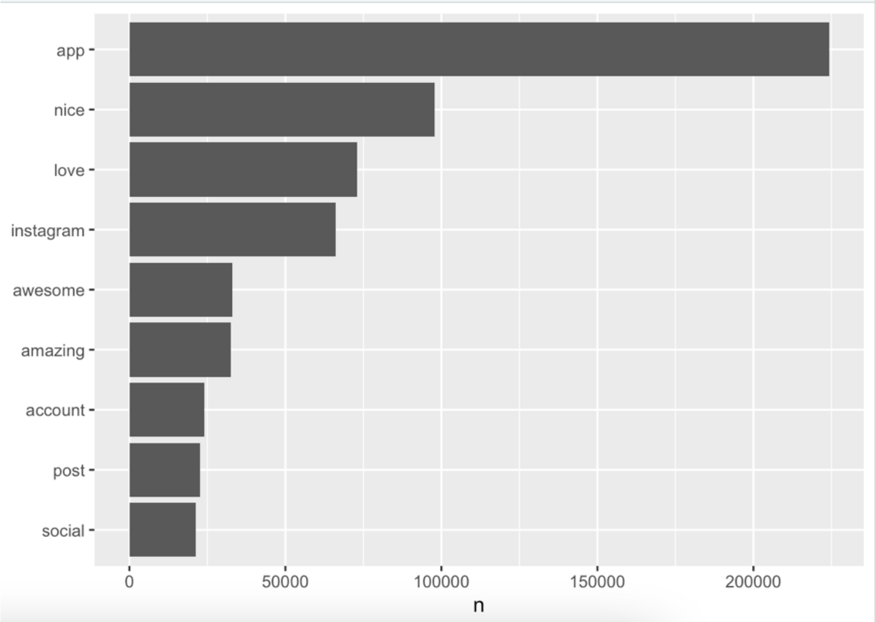
```
tidy_tt <- tt_df %>%
  unnest_tokens(word,text) %>%
  anti_join(stop_words)
```

```
#counting frequencies for tokens
tidy_tt %>%
  count(word, sort=TRUE)
```

4. PLOTTING TOKEN FREQUENCIES

```
#filter to show top 9-10 words

#INSTAGRAM
freq_hist_ig <-tidy_ig %>%
  count(word, sort=TRUE) %>%
  filter(n > 20000) %>%
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_hist_ig)
```

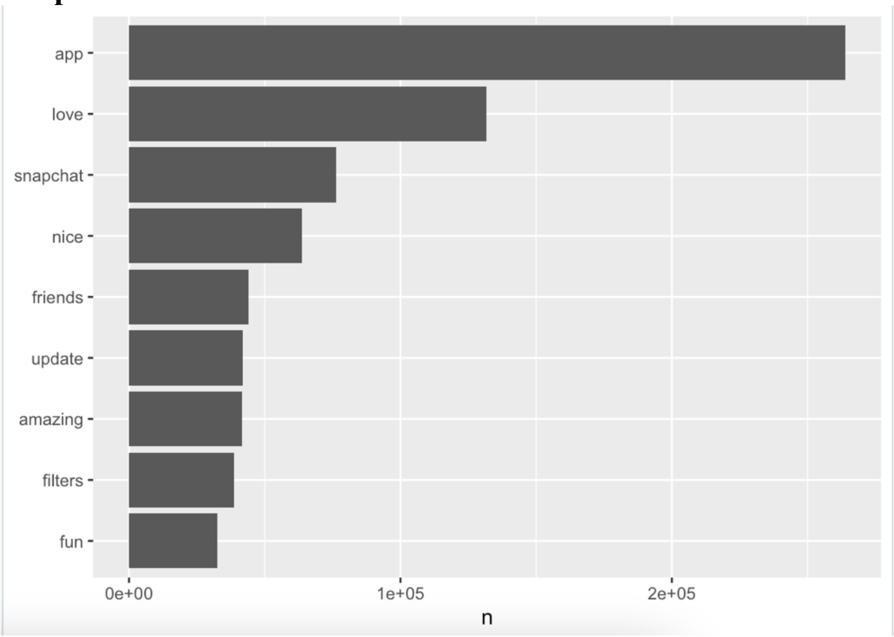
Output:



```
#SNAPCHAT
freq_hist_snap <-tidy_snap %>%
  count(word, sort=TRUE) %>%
  filter(n > 32000) %>%
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
```

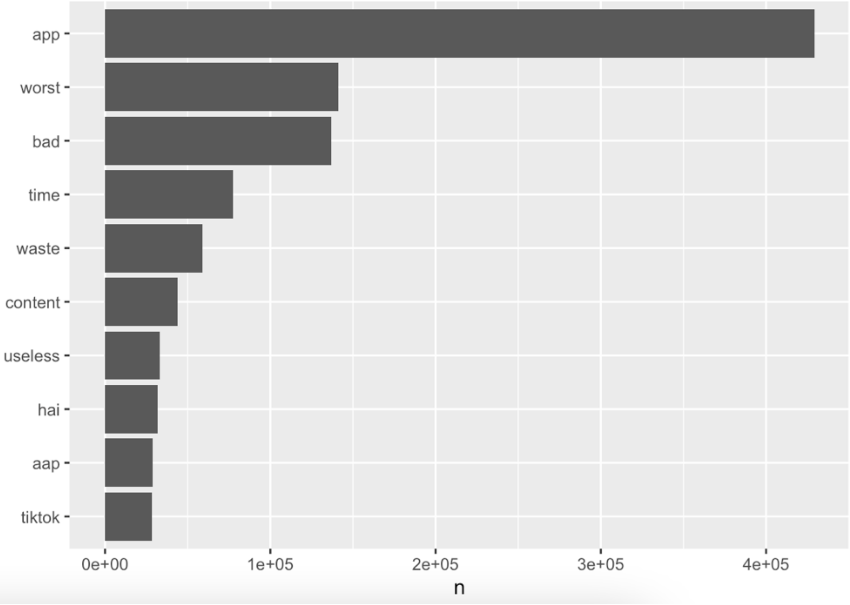
```
print(freq_hist_snap)
```

Output:



```
#TIKTOK
freq_hist_tt <-tidy_tt %>%
  count(word, sort=TRUE) %>%
  filter(n > 28000) %>%
  mutate(word = reorder(word,n )) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_hist_tt)
```

Output:



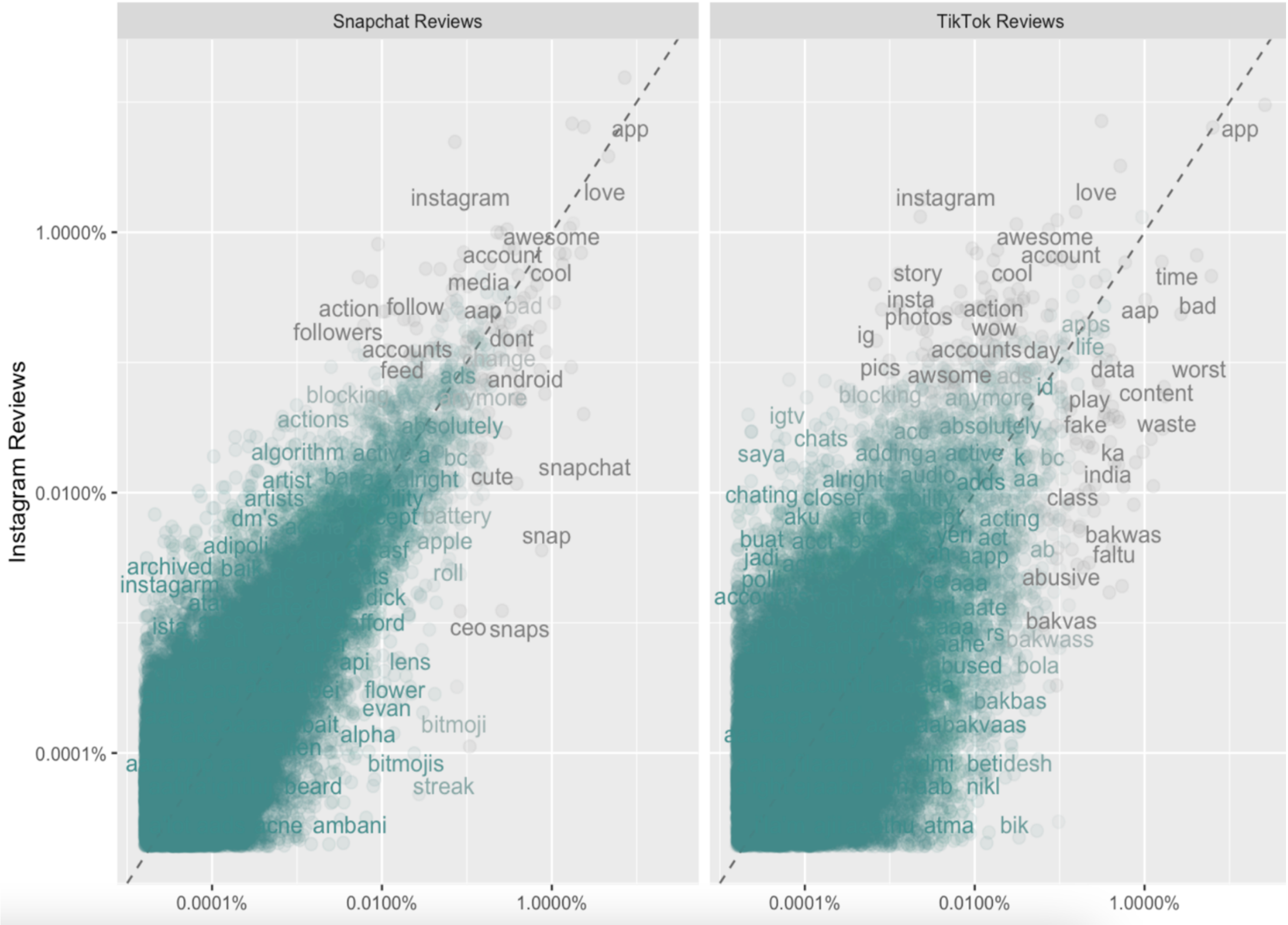
5. CORRELATION ANALYSIS

```
#concatenate all datasets for comparison
frequency <- bind_rows(mutate(tidy_ig, author="Instagram Reviews"),
                       mutate(tidy_snap, author= "Snapchat Reviews"),
                       mutate(tidy_tt, author="TikTok Reviews")) %>%
  mutate(word=str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n /sum(n))%>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion, `Snapchat Reviews`, `TikTok Reviews`)
```

5.1. CORRELOGRAMS

```
ggplot(frequency, aes(x=proportion, y=`Instagram Reviews`, color = abs(`Instagram Reviews`-
proportion)))+
  geom_abline(color="grey40", lty=2)+
  geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
  geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels= percent_format())+
  scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
  facet_wrap(~author, ncol=2)+
  theme(legend.position = "none")+
  labs(y= "Instagram Reviews", x=NULL)
```

Output:



5.2. CORRELATION TEST

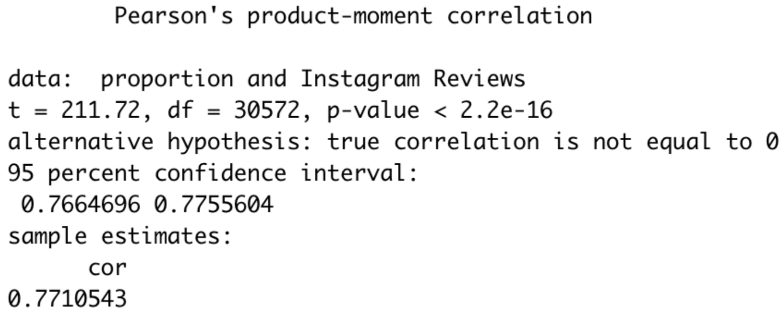
```
#INSTAGRAM x SNAPCHAT
ig_snap_corr <- cor.test(data=frequency[frequency$author == "Snapchat Reviews",],
        ~proportion + `Instagram Reviews`)
```

Output:

```
        Pearson's product-moment correlation

data:  proportion and Instagram Reviews
t = 390.41, df = 39987, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8879890 0.8920631
sample estimates:
      cor
0.8900438
```

```
#INSTAGRAM x TIKTOK
ig_tt_corr <- cor.test(data=frequency[frequency$author == "TikTok Reviews",],
        ~proportion + `Instagram Reviews`)
```

Output:



```
        Pearson's product-moment correlation

data:  proportion and Instagram Reviews
t = 211.72, df = 30572, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7664696 0.7755604
sample estimates:
      cor 
0.7710543 
```

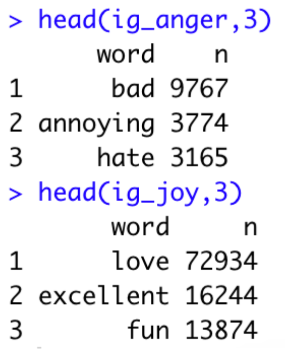
6. SENTIMENT ANALYSIS

```
#load 3 important libraries for sentiment analysis
afinn <- get_sentiments("afinn")
nrc <- get_sentiments("nrc")
bing <- get_sentiments("bing")

sentiment <- bind_rows(mutate(afinn, lexicon="afinn"),
                       mutate(nrc, lexicon= "nrc"),
                       mutate(bing, lexicon="bing"))
```

6.1. SENTIMENT LEXICON

```
#INSTAGRAM
#filter to show words with anger sentiment
nrcanger <- get_sentiments("nrc") %>%
           filter(sentiment == "anger")
#inner joining data and anger sentiments
ig_anger <- tidy_ig %>%
           inner_join(nrcanger) %>%
           count(word, sort=T)

#filter to show words with joy sentiment
nrcjoy <- get_sentiments("nrc") %>%
          filter(sentiment == "joy")
#inner joining data and joy sentiments
ig_joy <- tidy_ig %>%
          inner_join(nrcjoy) %>%
          count(word, sort=T)

#print results for comparison
head(ig_anger,3)
head(ig_joy,3)
```

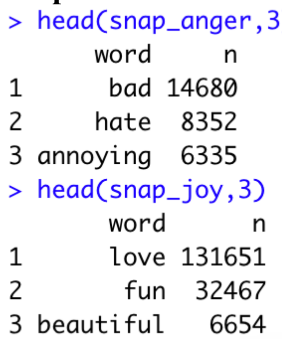
Output:



```
> head(ig_anger,3)
      word    n
1      bad 9767
2 annoying 3774
3     hate 3165
> head(ig_joy,3)
       word     n
1      love 72934
2 excellent 16244
3       fun 13874
```

```
#SNAPCHAT
#filter to show words with anger sentiment
nrcanger <- get_sentiments("nrc") %>%
  filter(sentiment == "anger")
#inner joining data and anger sentiments
snap_anger <- tidy_snap %>%
  inner_join(nrcanger) %>%
  count(word, sort=T)

#filter to show words with joy sentiment
nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
#inner joining data and joy sentiments
snap_joy <- tidy_snap %>%
  inner_join(nrcjoy) %>%
  count(word, sort=T)

#print results for comparison
head(snap_anger,3)
head(snap_joy,3)
```

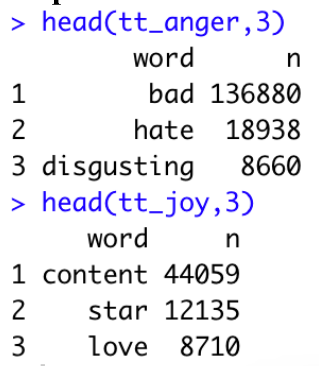
Output:



```
> head(snap_anger,3)
      word     n
1      bad 14680
2     hate  8352
3 annoying  6335
> head(snap_joy,3)
       word      n
1      love 131651
2       fun  32467
3 beautiful   6654
```

```
#TIKTOK
#filter to show words with anger sentiment
nrcanger <- get_sentiments("nrc") %>%
  filter(sentiment == "anger")
#inner joining data and anger sentiments
tt_anger <- tidy_tt %>%
  inner_join(nrcanger) %>%
  count(word, sort=T)

#filter to show words with joy sentiment
nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")
#inner joining data and joy sentiments
tt_joy <- tidy_tt %>%
  inner_join(nrcjoy) %>%
  count(word, sort=T)

#print results for comparison
head(tt_anger,3)
head(tt_joy,3)
```

Output:

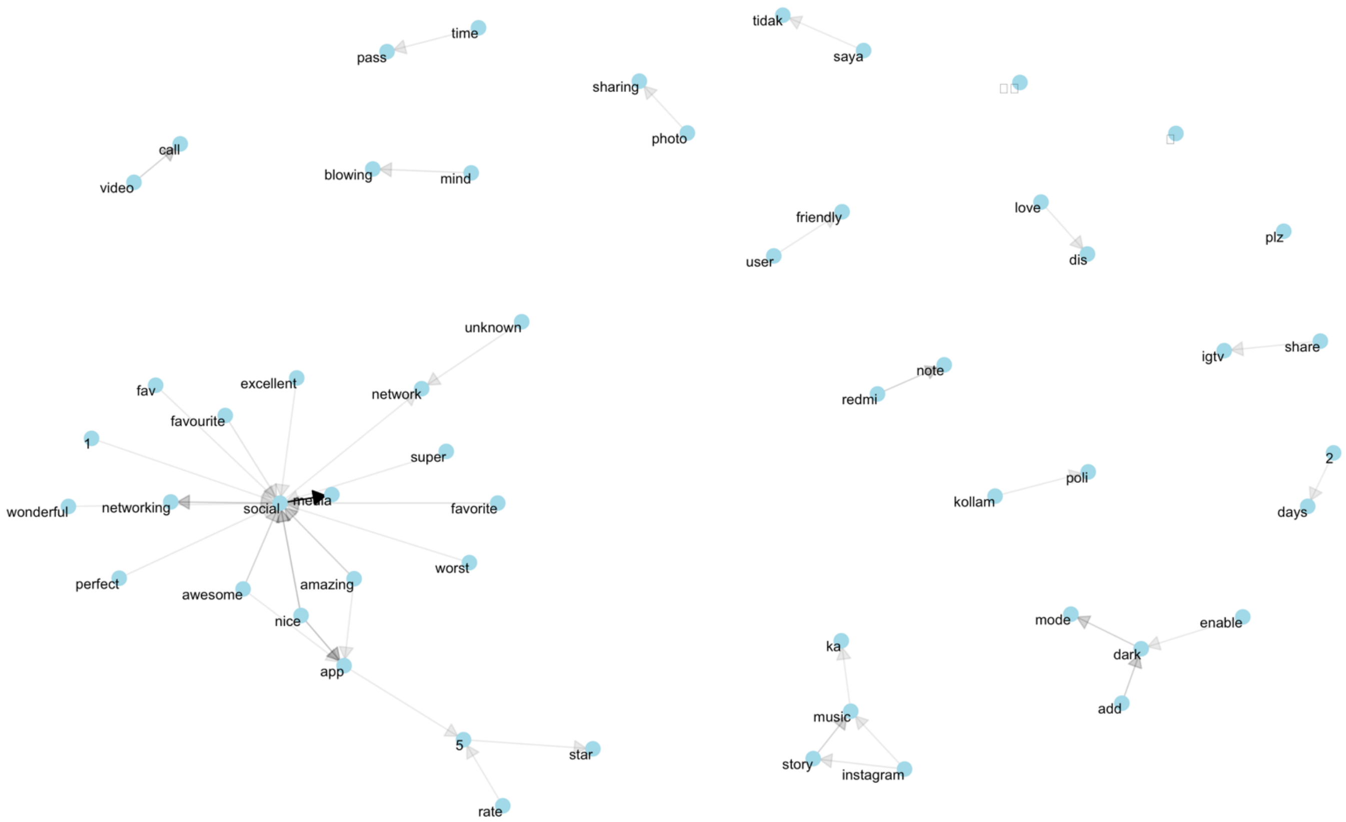


```
> head(tt_anger,3)
        word      n
1        bad 136880
2       hate  18938
3 disgusting   8660
> head(tt_joy,3)
     word     n
1 content 44059
2    star 12135
3    love  8710
```

6.2. TRIGRAM

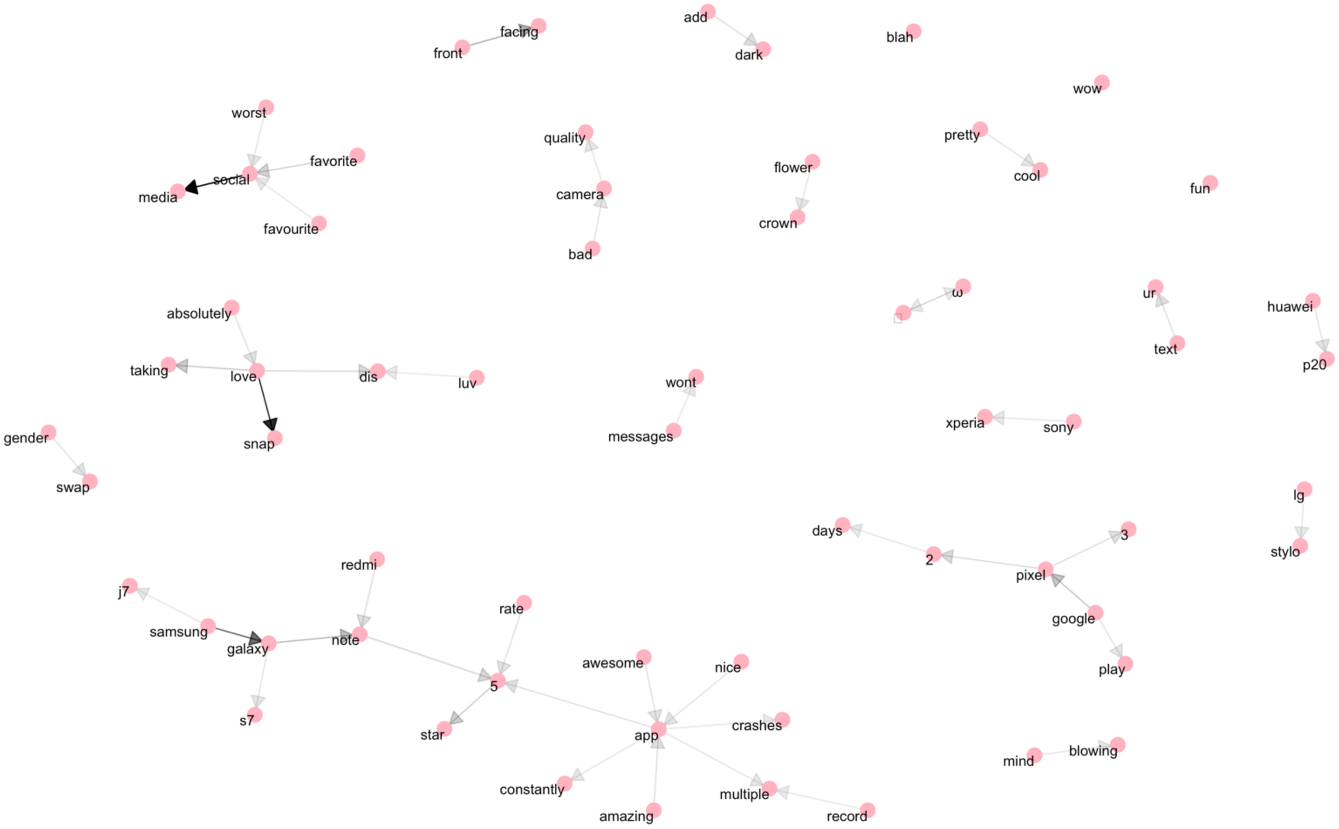
```
#INSTAGRAM
#prepare data; removing NA and stop words
trigram <- ig_df %>%
  unnest_tokens(trigram, text, token = "ngrams", n=3) %>%
  filter(!is.na(trigram))%>%
  separate(trigram, c("word1", "word2", "word3"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word)

#count trigram
trigram_counts <- trigram %>%
  count(word1, word2, word3, sort = TRUE)

#create matrix to draw trigram network
trigram_graph <- trigram_counts %>%
  filter(n>50) %>%
  graph_from_data_frame()

#visualize trigram network
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
ggraph(trigram_graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```

Output:



```
#SNAPCHAT
#prepare data; removing NA and stop words
trigram <- snap_df %>%
  unnest_tokens(trigram, text, token = "ngrams", n=3) %>%
  filter(!is.na(trigram))%>%
  separate(trigram, c("word1", "word2", "word3"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word)

#count trigram
trigram_counts <- trigram %>%
  count(word1, word2, word3, sort = TRUE)

#create matrix to draw trigram network
trigram_graph <- trigram_counts %>%
  filter(n>50) %>%
  graph_from_data_frame()

#visualize trigram network
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
ggraph(trigram_graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightpink", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```

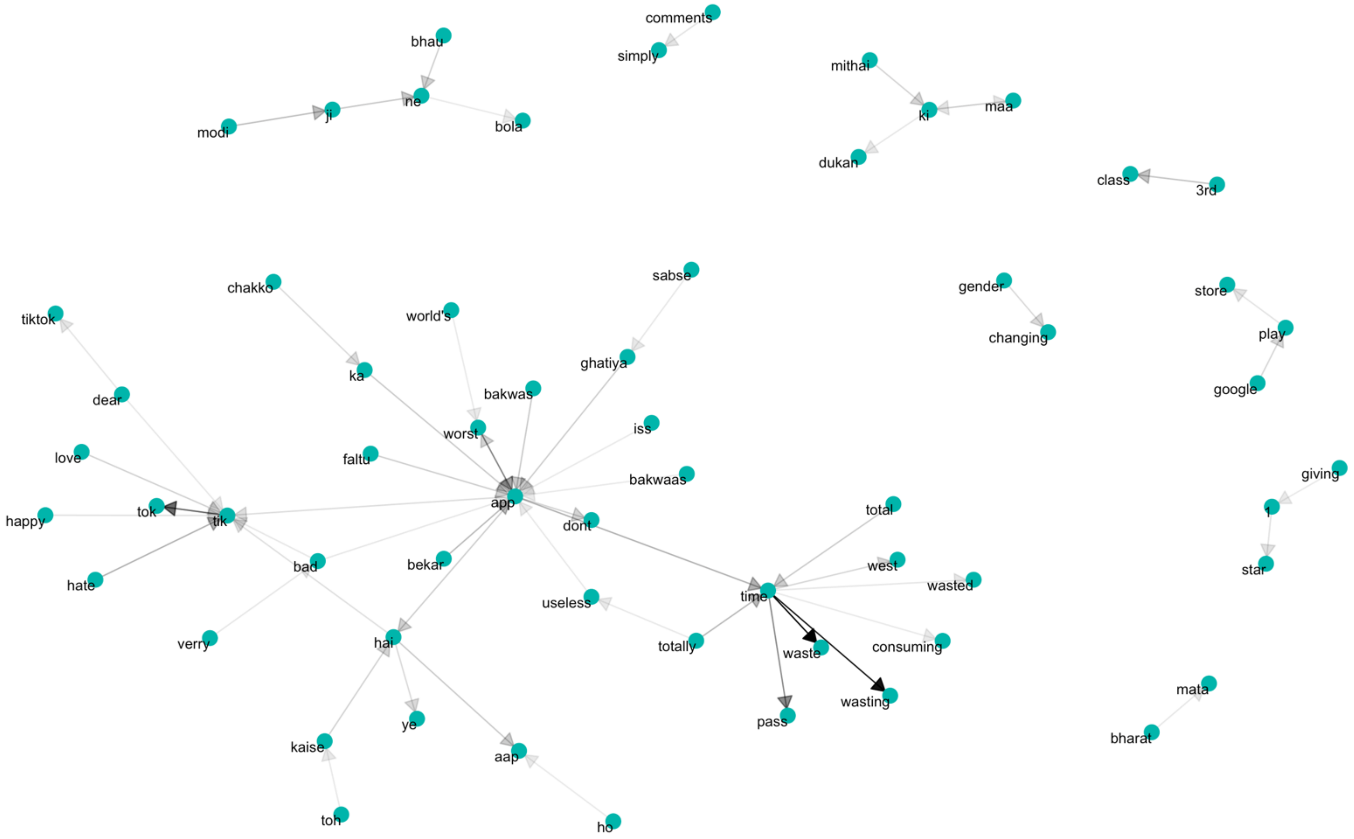
Output:



```
#TIKTOK
#prepare data; removing NA and stop words
trigram <- tt_df %>%
  unnest_tokens(trigram, text, token = "ngrams", n=3) %>%
  filter(!is.na(trigram))%>%
  separate(trigram, c("word1", "word2", "word3"), sep=" ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word)

#count trigram
trigram_counts <- trigram %>%
  count(word1, word2, word3, sort = TRUE)

#create matrix to draw trigram network
trigram_graph <- trigram_counts %>%
  filter(n>50) %>%
  graph_from_data_frame()

#visualize trigram network
```

```
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
ggraph(trigram_graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightpink", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```

Output:



comments
bhau
simply
mithai
maa
ne
ji
ki
bola
modi
dukan
class
3rd
sabse
chakko
gender
store
tiktok
world's
changing
play
ghatiya
ka
bakwas
google
dear
worst
iss
love
faltu
giving
bakwaas
1
app
total
happy
tok
tik
dont
hate
bad
bekar
west
time
wasted
star
useless
verry
hai
totally
waste
consuming
wasting
mata
pass
kaise
ye
aap
bharat
toh
ho