GLO TWITTER CUSTOMER CARE REPORT

ONIGBANJO MOWALOLA

08 February 2021

—

Text Analytics and Natural Language Processing (NLP) –

DAT-5317

—

Student ID - 9397115

TABLE OF CONTENTS

Contents

CASE BACKGROUND

1.1 ABOUT THE COMPANY

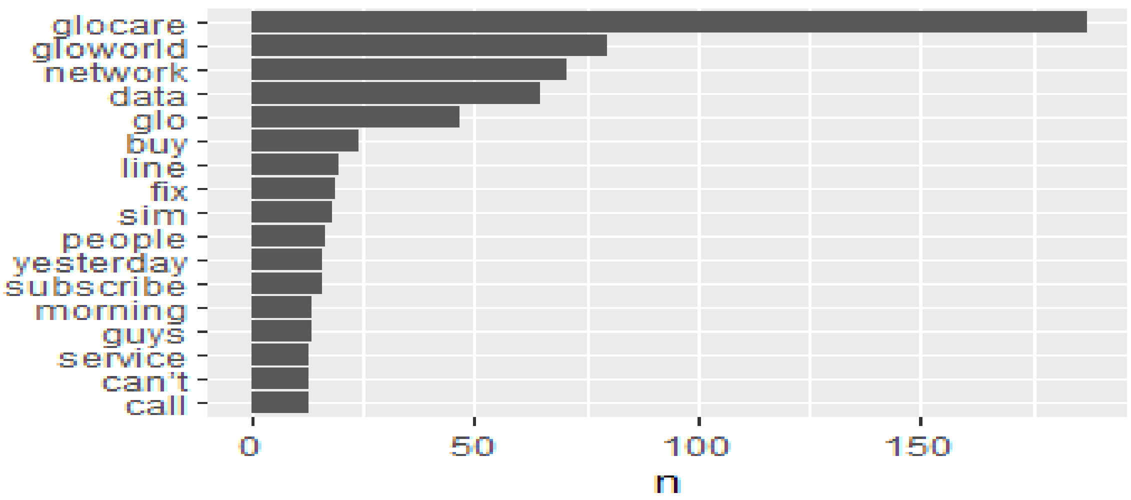
Globacom Nigeria Limited, popularly referred to by its users as Glo, is a Nigerian multinational telecommunications company, providing its users with mobile data plans, broadband, voice/sms plans, roaming, business/corporate packages and a host of other servces. As of December 2018, it had over 45, million subscribers on its network, making it the second largest network operator in Nigeria. Despite its success and many promotions, Glo has recently faced challenge in terms of meeting and even retaining its customers. There has been a spike in the number of complaints via its customer care helpdesk on Twitter. This data is assumed to be generated from mobile users only via the Twitter customer care page. (Globacom Limited, 2020)

I have been contacted as an independent analyst to review this and advise the company on the main areas of customer complaints, as well as recommend possible solutions to solve this.

1.2 INSIGHT FROM ANALYSIS

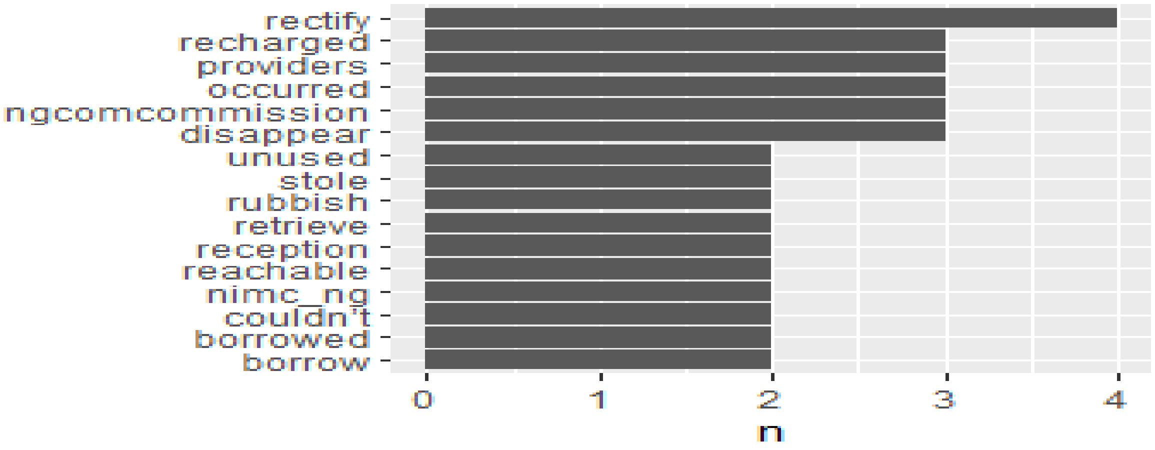
The data analysed was gotten from GloCare, the official customer care help line on Twitter. The data extracted has been scrubbed clean of replies to remove duplicated and unnecessary information.

From the chart below, the most common words are related to the name of the company which doesn't provide any insight at all.



The data therefore has to be cleaned further to get valuable insight.

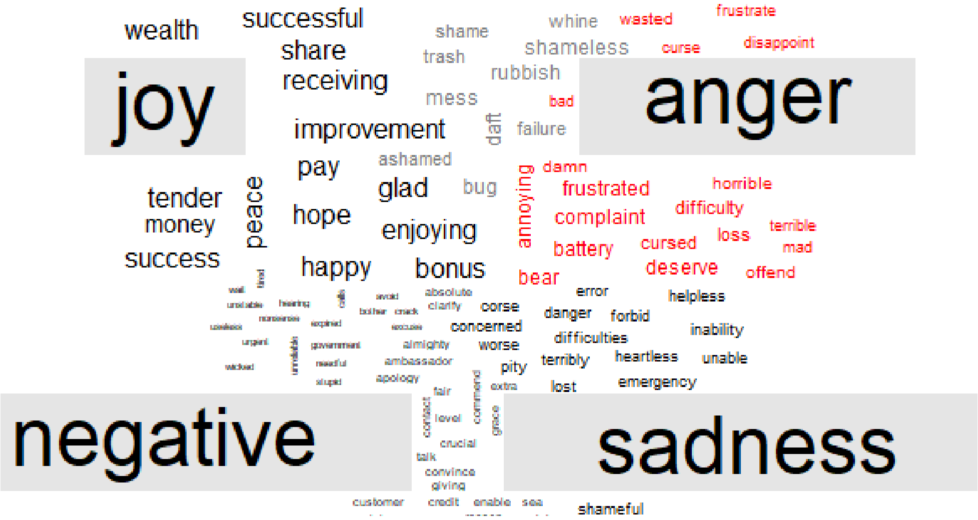
After cleaning the data, we can see that words like rectify, rubbish, couldn’t appear more frequently. However, this does not provide much insight because we are still unable to understand the nature of the customer complaints the account receives.



To understand customer sentiments based on the complaints in the tweets, we performed a sentiment analysis using the nrc lexicons as a framework for measuring emotions and perceptions. The nrc sentiment lexicons measure emotions in terms of positive, negative, anger, fear, anticipation, disgust, joy, surprise, sadness.

From the result of the analysis, we see that most of the words from the tweets are centered more around the anger, negative, sadness emotions. This shows that the tweets from the Glo users are more of complaints than they are recommendation or praises. This means that the company has more displeased users.

From the word clusters, we notice words like annoying, frustrate, loss, etc. There seems to be a general complaint about the service delivery of Glo, in terms of connectivity and network usage.

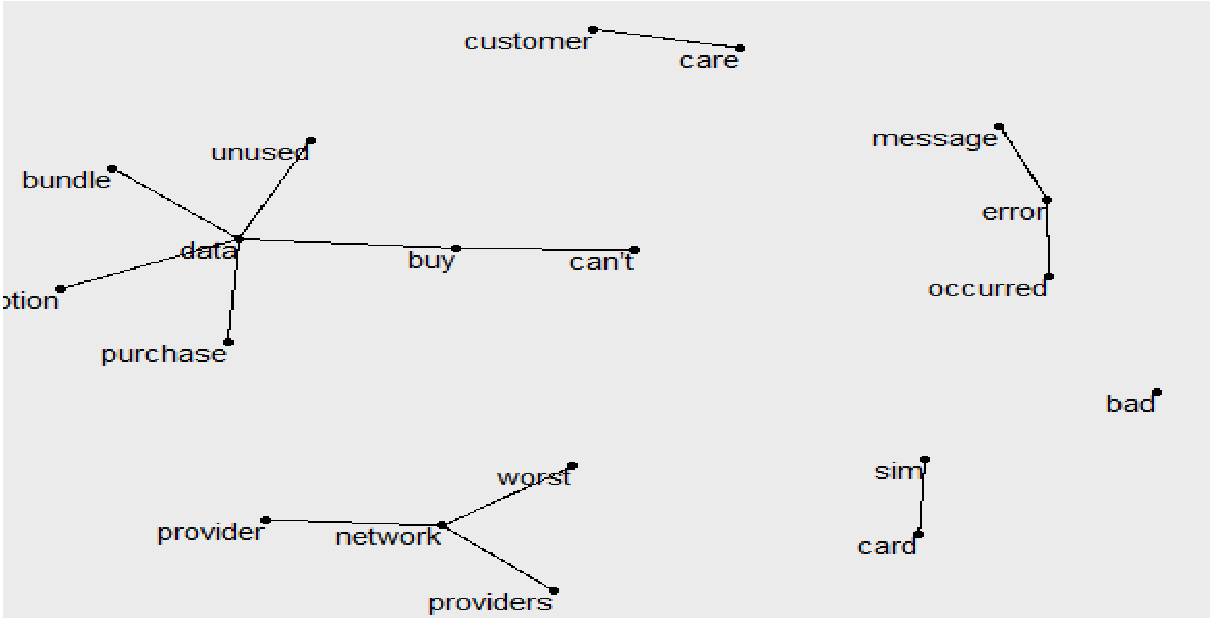


This analysis still does not give as much insight as needed. We can only conclude that the nature of the tweets on the GloCare page are mostly negative, sadness, and angry emotions which show the dissatisfaction of the users of the network.

Next, we perform an analysis on pairs of words to get more insight. From the image below, we notice a network of words which occur frequently on the page. We see the word data, has been linked to unused, can't, buy, bundle, indicating that a number of users face challenges when trying to purchase data bundles.

We understand that a lot of customers seem to experience an inability to purchase data plans. There also seems to be a lot of error messages being complained about. Users have problems with their sim cards. They seem to use the word bad a lot, indicating displeasure.

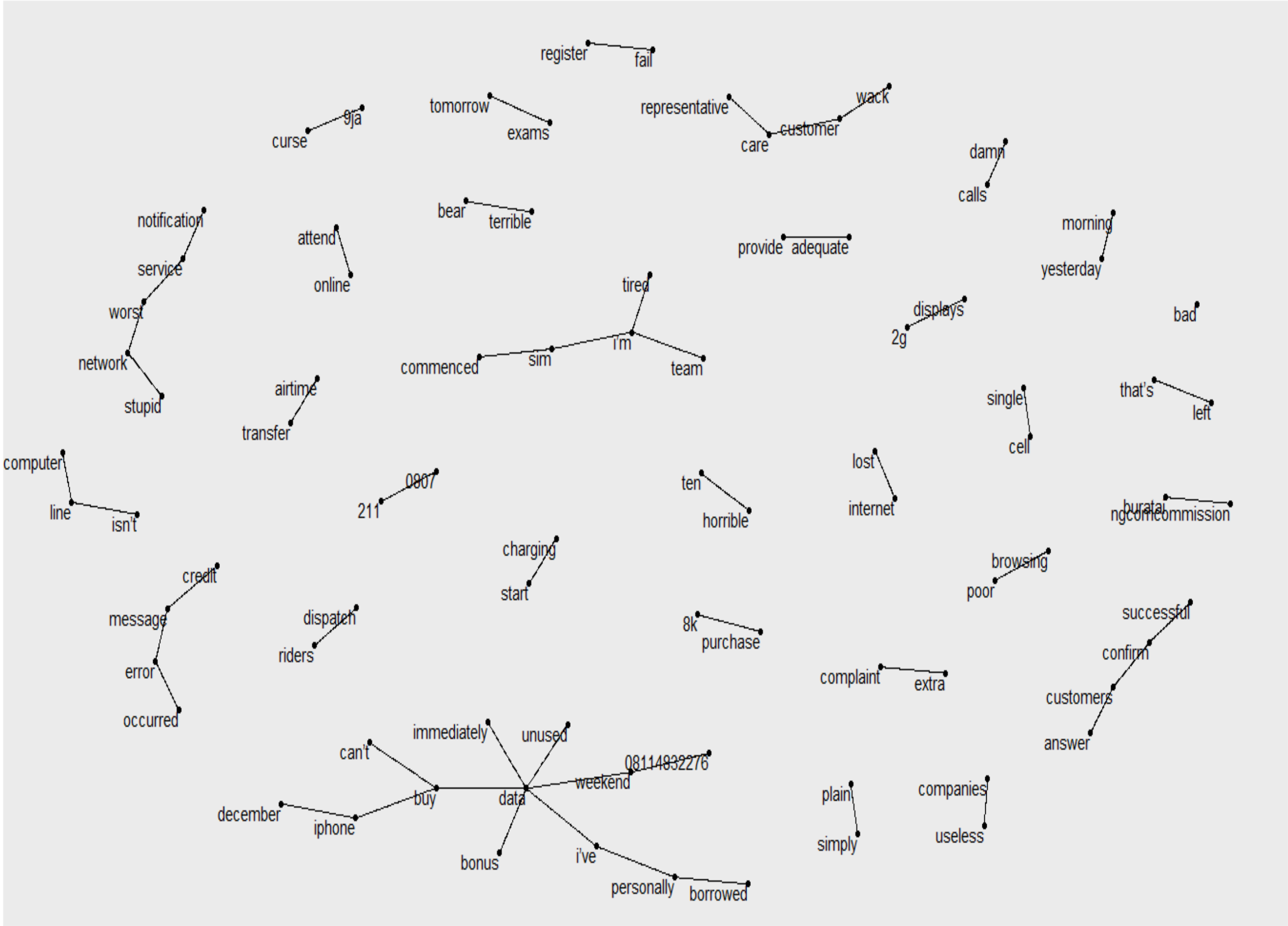
The most concerning link there would be the link that shows that a lot of the user complaints claim that the network provider is useless, the worst etc.



This shows that a lot of users are really displeased with the service of the Glo.

Further analysis on the word pairs when increased to 3 pairs of words, we see that complaints now revolve around failed sim registration, which could be as result of the new government regulations on sim and National Identification Number linkage.

There are also complaints relating to airtime transfer, unused data, notification, error messages, poor customer care representation, purchase and so on. Generally, we can see that there appears to be a lot of dissatisfaction from the customers using Glo network.



1.3 RECOMMENDATIONS

The network providers need to improve on their service by providing better facilities to its users to increase overall satisfaction gotten from using the network.

This can be achieved by regularly upgrading the network to ensure its users have a stronger and faster internet connection.

Seeing as a lot of users have problems purchasing bundles, Glo, needs to find easier means or its users to purchase bundles. This can be done by creating reminders and auto-renewal features for users. They can make their website more user friendly to make it easier for users to purchase or renew data bundles.

In line with new telecoms regulations, regarding linking National Identification Numbers(NIN) to sim cards, Glo can reduce the negative sentiments surrounding its products by dedicating a platform to collecting users NIN online. This would reduce foot traffic in its physical offices and reduce the stress of its users, thereby creating more positive sentiments for the company.

Glo can also improve on its response time to customer complaints, via the GloCare Twitter platform. It is advisable that Glo performs routine maintenance on its network, as well as participate in more customer surveys to find out how the network is performing. This would help the company to realize it weak areas before they become huge problems and advantage to its competitors.

Based on the results of the analysis, if the company fails to improve its services, more negative sentiments/feedback will be received from its customers, ultimately leading to an increase in the rate of dissatisfaction of its users, as well as reduction in revenue due to customers migrating to competing brands who can provide better services.

References

Globacom Limited. (2020). *About Us*. Retrieved from Glo: https://www.gloworld.com/ng/about-us/

APPENDIX

```
> ###########################################################################
>
> library(rtweet) #twitter library
> library(dplyr)  #pipes
> library(stringr)
> library(tidyr)
> library(tidytext)
> library(textdata)
> library(ggplot2)
> library(wordcloud)
> library(reshape2)
> library(tm)
> library(ggplot2)
> library(igraph)
> library(ggraph)
>
> # Running a search using the keywords "GloCare to get tweets by customers using
> # Glo mobile network
> feedback <- search_tweets(
+   "GloCare -filter:replies", n = 18000, include_rts = FALSE, lang = "en"
+ )
Downloading [>---------------------------------------------]   2%>
> #Process tweets into tidy text/corpus objects
> raw_df <- feedback %>% select(screen_name,text)
>
> #Manually removing http elememts from data
> raw_df$text <- gsub("http\\S+","", raw_df$text)
>
>
> #Tokenizing and counting the frequency of words
> glo_df <- raw_df %>%
+   unnest_tokens(word,text) %>%
+   anti_join(stop_words) %>%
+   count(word, sort=TRUE) %>%
+   mutate(word=reorder(word, n))
Joining, by = "word"
>
> print(glo_df)
# A tibble: 728 x 2
   word         n
   <fct>    <int>
 1 glocare    187
 2 gloworld    80
 3 network     71
 4 data        65
 5 glo         47
 6 buy         24
 7 line        20
 8 fix         19
 9 sim         18
10 people      17
# ... with 718 more rows
>
> #Removing numbers from data as they don't provide any useful insight
> glo_df <- glo_df[-grep("\\b\\d+\\b",glo_df$word),]
>
> #Plotting to visualize the most common words in the data
> df_plot<- glo_df %>%
+   top_n(15)                   %>%
+   ggplot(aes(word, n))+
+   geom_col()+
+   xlab(NULL)+
+   coord_flip()
Selecting by n
>
> print(df_plot)
>
> # calling the stop words
> data("stop_words")
>
>
> #Creating new stop words for better analysis
> junk <- data_frame(lexicon = c("CUSTOM"),
+                   word=c("abeg", "ẹ", "ọmọ", "dear","kindly","can't","guys", "ụnụ",
+                          "9mobile","glocare", "gloworld", "glo","i'm","dey","una",
+                          "people", "happening", "time", "days","wrong","plan","past",
+                          "https", "url", "t.co", "airtelnigeria", "na", "shit", "don't",
+                          "glonigeria", "pls","y'all","abi","don","i've", "mtnng","omo",
+                          "ooo","wtf","4g","airtel","cc","consumersncc", "amp","what's",
+                        "mtn180","wahala","ekiti","abeokuta","haba","ehn","friend","god",
+                        "yo", "dms","cos","fuck","gan","fit","hrs","ashawo","3bood021",
+                        "aramoko","anddddd","anuofia","awka","be4","awoof","biko","benin",
+                        "buhari","burutai","choima","dakun","da","de","dai","donnn",
+                        "dnt","efcc","egbu","egwugwu","ehnnnn","fccpcnigeria","fash360degree",
+                        "fgs","ffs","globalcom","glooo","glonetworks","gloneedtotalrevival",
+                        "hsadam8","hey","ibadan","i'll","igbu","igpadamu","ijoku","jésù",
+                        "inshallah","kain","lemme","kiloshele","lagos","lautech","mehnnn",
+                        "mikeadenuga_glo","mi","mk","motherfuxkers","mtn","msg","mumu",
+                        "n1000","n16","naijapatriotreport","n400","n500","naaa","ntori",
+                        "oautwitter","ogungokillu","ohhh","ole","olumupa","oky","olohun",
+                        "olosho","omoogun","omor","oo","oshi","papa","puh_lease","shey",
+                        "proudlyfromijebuigbo","tejubabyfacetv","tf","tiktok_us","wa","wah",
+                        "wayy","werey","wetin","why's","wizkid","ya","you're","i've","robber",
+                        "armed","24","hours","777","calling","1gb","one's","700","ive","67mb"))
>
> # Creating a new list of stop words with custom stop words and existing stop words
> stop_words_full <- rbind(stop_words,junk)
>
> # Plotting to see the most common words used in the data
> freq_df<- glo_df %>%
+   anti_join(stop_words_full) %>%
+   anti_join(stopwordslangs) %>%
+   #count(word, sort=TRUE) %>%
+   mutate(word=reorder(word, n)) %>%
+   filter(n < 5) %>%
+   top_n(15)                  %>%
+   ggplot(aes(word, n))+
+   geom_col()+
+   xlab(NULL)+
+   coord_flip()
Joining, by = "word"
Joining, by = "word"
Selecting by n
>
> print(freq_df)
>
>
> # Running sentiment analysis for Words used
> glo_sentiments <- glo_df %>%
+   anti_join(stopwordslangs) %>%
+   inner_join(get_sentiments("nrc")) %>%
+   filter(sentiment %in% c("positive", "anger","negative","disgust","sadness","joy")) %>%
+   count(word, sentiment, sort=T)     %>%
+   ungroup() %>%
+   mutate(word=reorder(word, n))
```

```
Joining, by = "word"
Joining, by = "word"
>
> print(glo_sentiments)
# A tibble: 29 x 3
   word       sentiment     n
   <fct>      <chr>     <int>
 1 clarify    positive      1
 2 commend    positive      1
 3 daft       disgust       1
 4 daft       negative      1
 5 difficulty anger         1
 6 difficulty negative      1
 7 difficulty sadness       1
 8 enable     positive      1
 9 frustrate  anger         1
10 frustrate  disgust       1
# ... with 19 more rows
>
> # Creating a wordcloud to view nrc lexicon word associations with coomon words from glo_df
> glo_wordcloud <-   glo_df %>%
+     anti_join(stop_words_full) %>%
+     inner_join(get_sentiments("nrc")) %>%
+     filter(sentiment %in% c("positive", "anger","negative","disgust","sadness","joy")) %>%
+     count(word, sentiment, sort=TRUE) %>%
+     ungroup() %>%
+     acast(word ~sentiment, value.var="n", fill=0) %>%
+     comparison.cloud(colors = c("red", "grey50"),
+                      max.words=10000, scale = c(1,0.2))
Joining, by = "word"
Joining, by = "word"
Warning messages:
1: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  cooperation could not be fit on page. It will not be plotted.
2: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  prepare could not be fit on page. It will not be plotted.
3: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  real could not be fit on page. It will not be plotted.
4: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  received could not be fit on page. It will not be plotted.
5: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  rectify could not be fit on page. It will not be plotted.
6: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  renewal could not be fit on page. It will not be plotted.
7: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  university could not be fit on page. It will not be plotted.
```

```
8: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  unlimited could not be fit on page. It will not be plotted.
9: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  worth could not be fit on page. It will not be plotted.
10: In comparison.cloud(., colors = c("red", "grey50"), max.words = 10000,  :
  immediately could not be fit on page. It will not be plotted.
>
> ##############################################
> ###### N-grams and tokenizing ###############
> ##############################################
>
> # Creating bigrams using unfiltered data
> glo_bigrams <- feedback %>%
+   unnest_tokens(bigram, text, token = "ngrams", n = 2)
>
> glo_bigrams #We want to see the bigrams (words that appear together, "pairs")
# A tibble: 4,053 x 90
   user_id status_id created_at          screen_name source display_text_wi~ reply_to_status~
   <chr>   <chr>     <dttm>              <chr>       <chr>             <dbl> <lgl>
 1 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 2 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 3 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 4 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 5 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 6 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 7 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 8 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 9 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
10 295913~ 13595429~ 2021-02-10 16:41:08 SaleemIOth~ Tweet~               82 NA
# ... with 4,043 more rows, and 83 more variables: reply_to_user_id <lgl>,
#   reply_to_screen_name <lgl>, is_quote <lgl>, is_retweet <lgl>, favorite_count <int>,
#   retweet_count <int>, quote_count <int>, reply_count <int>, hashtags <list>, symbols <list>,
#   urls_url <list>, urls_t.co <list>, urls_expanded_url <list>, media_url <list>,
#   media_t.co <list>, media_expanded_url <list>, media_type <list>, ext_media_url <list>,
#   ext_media_t.co <list>, ext_media_expanded_url <list>, ext_media_type <chr>,
#   mentions_user_id <list>, mentions_screen_name <list>, lang <chr>, quoted_status_id <chr>,
#   quoted_text <chr>, quoted_created_at <dttm>, quoted_source <chr>,
#   quoted_favorite_count <int>, quoted_retweet_count <int>, quoted_user_id <chr>,
#   quoted_screen_name <chr>, quoted_name <chr>, quoted_followers_count <int>,
#   quoted_friends_count <int>, quoted_statuses_count <int>, quoted_location <chr>,
#   quoted_description <chr>, quoted_verified <lgl>, retweet_status_id <chr>,
#   retweet_text <chr>, retweet_created_at <dttm>, retweet_source <chr>,
#   retweet_favorite_count <int>, retweet_retweet_count <int>, retweet_user_id <chr>,
#   retweet_screen_name <chr>, retweet_name <chr>, retweet_followers_count <int>,
#   retweet_friends_count <int>, retweet_statuses_count <int>, retweet_location <chr>,
#   retweet_description <chr>, retweet_verified <lgl>, place_url <chr>, place_name <chr>,
#   place_full_name <chr>, place_type <chr>, country <chr>, country_code <chr>,
#   geo_coords <list>, coords_coords <list>, bbox_coords <list>, status_url <chr>, name <chr>,
#   location <chr>, description <chr>, url <chr>, protected <lgl>, followers_count <int>,
#   friends_count <int>, listed_count <int>, statuses_count <int>, favourites_count <int>,
#   account_created_at <dttm>, verified <lgl>, profile_url <chr>, profile_expanded_url <chr>,
#   account_lang <lgl>, profile_banner_url <chr>, profile_background_url <chr>,
#   profile_image_url <chr>, bigram <chr>
>
> glo_bigrams %>%
+   count(bigram, sort = TRUE) #this has many stop words, need to remove them
# A tibble: 3,069 x 2
   bigram               n
   <chr>            <int>
 1 gloworld glocare    42
 2 https t.co          34
 3 glocare gloworld    26
 4 buy data            21
 5 i have              18
 6 your network        17
 7 since yesterday     13
 8 you people          13
 9 has been            12
10 you guys            12
# ... with 3,059 more rows
>
> #to remove stop words from the bigram data, we need to use the separate function:
>
> bigrams_separated <- glo_bigrams %>%
+   separate(bigram, c("word1", "word2"), sep = " ")
>
> bigrams_filtered <- bigrams_separated %>%
+   filter(!word1 %in% stop_words_full$word) %>%
+   filter(!word2 %in% stop_words_full$word)%>%
+   filter(!word1 == "NA")%>%
+   filter(!word2 =="NA")
>
> #creating the new bigram, "no-stop-words":
> bigram_counts <- bigrams_filtered %>%
+   count(word1, word2, sort = TRUE)
> #want to see the new bigrams
> bigram_counts
# A tibble: 227 x 3
   word1   word2          n
   <chr>   <chr>      <int>
 1 buy     data          21
 2 can’t   buy            7
 3 customer care           4
 4 error    message        4
 5 network  provider       4
 6 sim      card           3
 7 bad      bad            2
 8 data     bundle         2
 9 data     subscription   2
10 error    occurred       2
# ... with 217 more rows
>
> bigram_graph <- bigram_counts %>%
+   filter(n>1) %>%
+   ungroup() %>%
+   graph_from_data_frame()
>
> bigram_graph
IGRAPH 8cc8b56 DN-- 19 14 --
+ attr: name (v/c), n (e/n)
+ edges from 8cc8b56 (vertex names):
 [1] buy     ->data         can't   ->buy          customer->care         error   ->message
 [5] network ->provider     sim     ->card         bad     ->bad          data    ->bundle
 [9] data    ->subscription error   ->occurred     network ->providers    purchase->data
[13] unused  ->data         worst   ->network
>
>
> ggraph(bigram_graph, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
>
>
> # Creating trigrams
>
> glo_trigram <- feedback %>%
+   unnest_tokens(trigram, text, token = "ngrams", n = 3)
>
> glo_trigram #We want to see the trigram (words that appear together, "pairs")
# A tibble: 3,856 x 90
   user_id status_id created_at          screen_name source display_text_wi~ reply_to_status~
   <chr>   <chr>     <dttm>              <chr>       <chr>             <dbl> <lgl>
 1 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 2 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 3 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 4 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 5 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 6 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~               40 NA
 7 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~              40 NA
 8 295913~ 13596110~ 2021-02-10 21:11:57 SaleemIOth~ Tweet~              40 NA
 9 295913~ 13595429~ 2021-02-10 16:41:08 SaleemIOth~ Tweet~              82 NA
10 295913~ 13595429~ 2021-02-10 16:41:08 SaleemIOth~ Tweet~              82 NA
# ... with 3,846 more rows, and 83 more variables: reply_to_user_id <lgl>,
#   reply_to_screen_name <lgl>, is_quote <lgl>, is_retweet <lgl>, favorite_count <int>,
#   retweet_count <int>, quote_count <int>, reply_count <int>, hashtags <list>, symbols <list>,
#   urls_url <list>, urls_t.co <list>, urls_expanded_url <list>, media_url <list>,
#   media_t.co <list>, media_expanded_url <list>, media_type <list>, ext_media_url <list>,
#   ext_media_t.co <list>, ext_media_expanded_url <list>, ext_media_type <chr>,
#   mentions_user_id <list>, mentions_screen_name <list>, lang <chr>, quoted_status_id <chr>,
#   quoted_text <chr>, quoted_created_at <dttm>, quoted_source <chr>,
#   quoted_favorite_count <int>, quoted_retweet_count <int>, quoted_user_id <chr>,
#   quoted_screen_name <chr>, quoted_name <chr>, quoted_followers_count <int>,
#   quoted_friends_count <int>, quoted_statuses_count <int>, quoted_location <chr>,
#   quoted_description <chr>, quoted_verified <lgl>, retweet_status_id <chr>,
#   retweet_text <chr>, retweet_created_at <dttm>, retweet_source <chr>,
#   retweet_favorite_count <int>, retweet_retweet_count <int>, retweet_user_id <chr>,
#   retweet_screen_name <chr>, retweet_name <chr>, retweet_followers_count <int>,
#   retweet_friends_count <int>, retweet_statuses_count <int>, retweet_location <chr>,
#   retweet_description <chr>, retweet_verified <lgl>, place_url <chr>, place_name <chr>,
#   place_full_name <chr>, place_type <chr>, country <chr>, country_code <chr>,
#   geo_coords <list>, coords_coords <list>, bbox_coords <list>, status_url <chr>, name <chr>,
#   location <chr>, description <chr>, url <chr>, protected <lgl>, followers_count <int>,
#   friends_count <int>, listed_count <int>, statuses_count <int>, favourites_count <int>,
#   account_created_at <dttm>, verified <lgl>, profile_url <chr>, profile_expanded_url <chr>,
#   account_lang <lgl>, profile_banner_url <chr>, profile_background_url <chr>,
#   profile_image_url <chr>, trigram <chr>
>
> glo_trigram %>%
+   count(trigram, sort = TRUE) #this has many stop words, need to remove them
# A tibble: 3,623 x 2
   trigram                n
   <chr>              <int>
 1 can’t buy data         7
 2 to buy data            7
 3 been able to           6
 4 i can’t buy            6
 5 my glo line            6
 6 for the past           5
 7 able to buy            4
 8 been trying to         4
 9 fix your network       4
10 glocare https t.co     4
# ... with 3,613 more rows
>
```

```
> #to remove stop words from the trigram data, we need to use the separate function:
>
> trigram_separated <- glo_trigram %>%
+   separate(trigram, c("word1", "word2","word3"), sep = " ")
>
> trigrams_filtered <- trigram_separated %>%
+   filter(!word1 %in% stop_words_full$word) %>%
+   filter(!word2 %in% stop_words_full$word) %>%
+   filter(!word3 %in% stop_words_full$word) %>%
+   filter(!word1 %in% NA) %>%
+   filter(!word2 %in% NA) %>%
+   filter(!word3 %in% NA)
>
> #creating the new bigram, "no-stop-words":
> trigram_counts <- trigrams_filtered %>%
+   count(word1, word2, word3, sort = TRUE)
>
> #want to see the new trigrams
> trigram_counts
# A tibble: 56 x 4
   word1       word2     word3       n
   <chr>       <chr>     <chr>   <int>
 1 can't       buy       data        7
 2 0807        211       6420        1
 3 08114832276 weekend   plans       1
 4 answer      customers phone       1
 5 attend      online    classes     1
 6 bad         bad       bad         1
 7 bear        terrible  networks    1
 8 bonus       data      worth       1
 9 buy         data      ahahn       1
10 buy         data      i've        1
# ... with 46 more rows
>
>
> trigram_graph <-trigram_counts %>%
+   filter(n > 0) %>%
+   graph_from_data_frame()
>
> trigram_graph
IGRAPH 8e95d26 DN-- 85 56 --
+ attr: name (v/c), word3 (e/c), n (e/n)
+ edges from 8e95d26 (vertex names):
 [1] can't      ->buy         0807       ->211         08114832276->weekend
 [4] answer     ->customers   attend     ->online      bad        ->bad
 [7] bear       ->terrible    bonus      ->data        buy        ->data
```

```
[10] buy        ->data           buy        ->iphone         care       ->representative
[13] commenced  ->sim            complaint  ->extra          confirm    ->successful
[16] curse      ->9ja            customer   ->care           customers  ->confirm
[19] damn       ->calls          data       ->i’ve           data       ->immediately
[22] december   ->iphone         dispatch   ->riders         displays   ->2g
+ ... omitted several edges
>
> ggraph(trigram_graph, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
> trigram_graph <-trigram_counts %>%
+   filter(n > 0) %>%
+   ungroup() %>%
+   graph_from_data_frame()
>
> trigram_graph
IGRAPH afd17af DN-- 85 56 --
+ attr: name (v/c), word3 (e/c), n (e/n)
+ edges from afd17af (vertex names):
 [1] can’t      ->buy            0807       ->211            08114832276->weekend
 [4] answer     ->customers      attend     ->online         bad        ->bad
 [7] bear       ->terrible       bonus      ->data           buy        ->data
[10] buy        ->data           buy        ->iphone         care       ->representative
[13] commenced  ->sim            complaint  ->extra          confirm    ->successful
[16] curse      ->9ja            customer   ->care           customers  ->confirm
[19] damn       ->calls          data       ->i’ve           data       ->immediately
[22] december   ->iphone         dispatch   ->riders         displays   ->2g
+ ... omitted several edges
>
> ggraph(trigram_graph, layout = "fr") +
+   geom_edge_link()+
+   geom_node_point()+
+   geom_node_text(aes(label=name), vjust =1, hjust=1)
>
```