UNIVERSITY OF WESTMINSTER

Name: Bhaumikkumar Patel

Student ID: w1813148

7BUIS025W

Web and Social Media Analytics

Coursework: Social Media Assignment (2020/21)

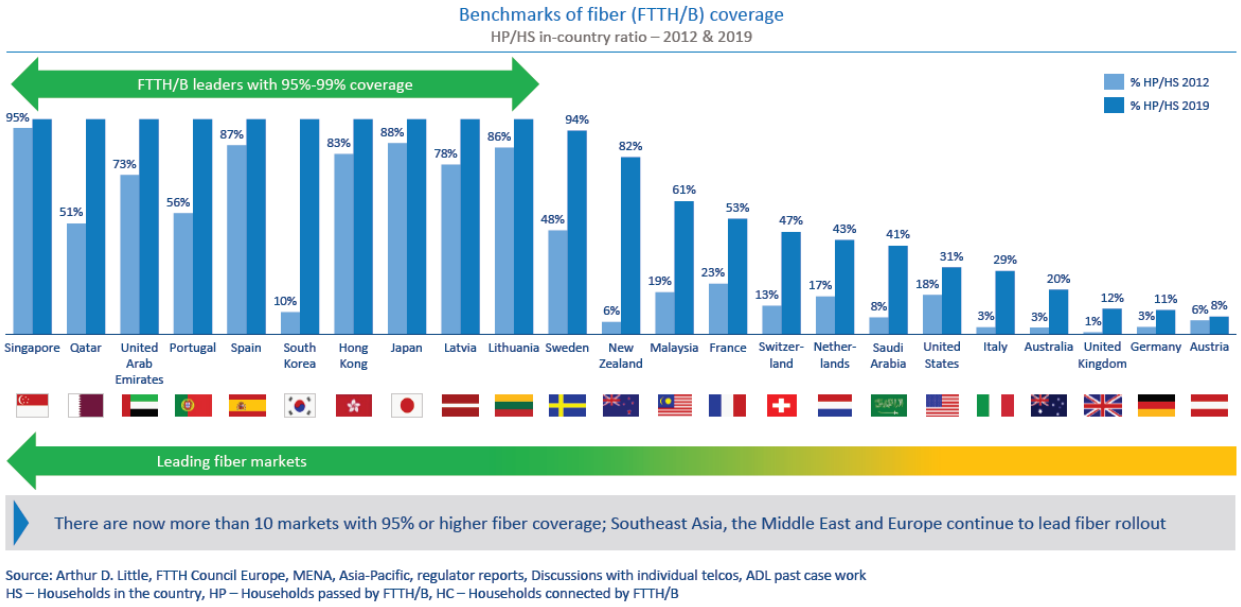
WORD COUNT: 1244

Identify a brand or famous person of interest and create a list of 5 possible keywords you could use to identify relevant tweets from Twitter.

I used the main keyword related to the Optical Fiber (Fibre in British English), it is a transparent and flexible made by glass of silica or plastic. Its diameter like slightly thicker than human hair. Optical Fibre are used for Telecom communication were transmitting over long distance as well as get higher bandwidth data transfer then electric cables. Fibers are sent data signals from one end to other end with less loss compared to the metal wires.

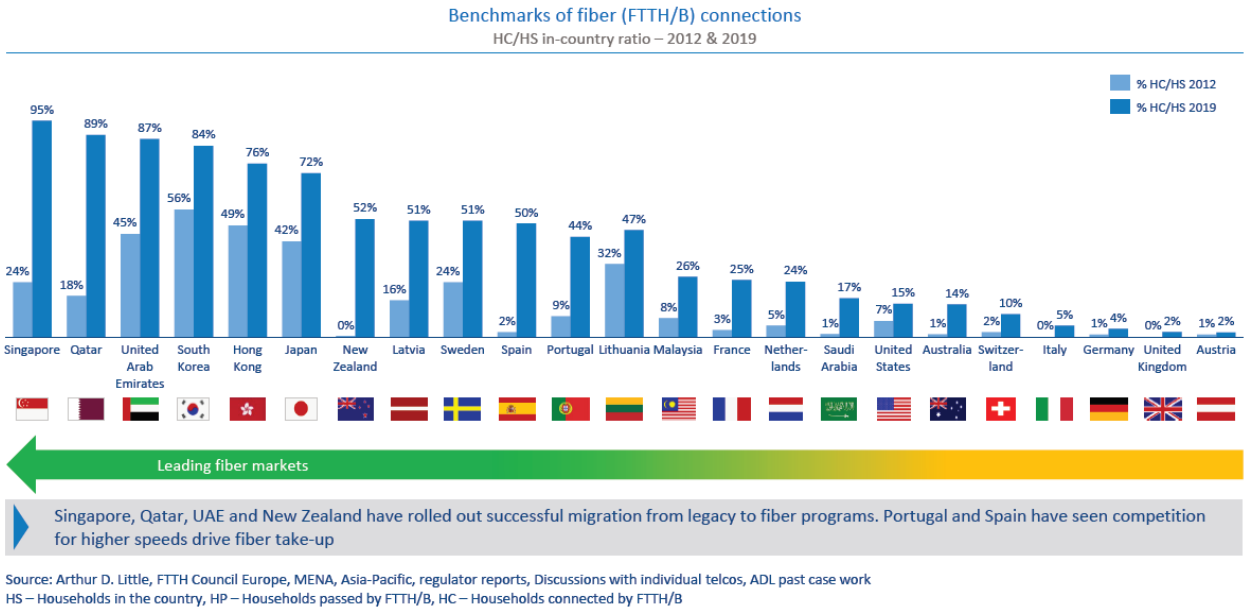
The first Optical fibre data transmission system demonstrated by German physicist Manfred Borner at Telefunken Research Labs in Ulm in 1965. The latest data transmission over fibre optic cable is 500 Gbit/s. [1] From around the world there are with 95% or higher fibre coverage; southeast Asia, the middle east and Europe continue to lead fibre rollout. On the other side Singapore, Qatar, UAE and New Zealand have rolled out successful migration. Portugal and Spain have seen competition for higher speeds drive fibre take up. Below figure explain the coverage and connection from various countries.

Figure 1. FTTH/B households Passed [2]



Source: Arthur D. Little, FTTH Council Europe, MENA, Asia-Pacific, regulator reports, Discussions with individual telcos, ADL past case work
HS – Households in the country, HP – Households passed by FTTH/B, HC – Households connected by FTTH/B

Figure 2. Fibre (FTTH/B) households connected [2]



I used five keywords as follows “fibre”, “fiberinternet”, “Internet”, “Broadband”, “Gigabyte”. These keywords are used by the service providers to explain new technology to the customer as well as the technicians tweets their working ethics.

Explain what is meant by an API and compare and contrast the two data collection APIs available on the Twitter platform.

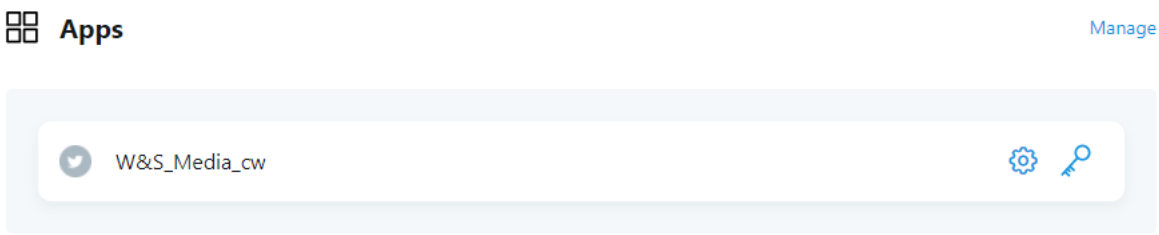
The API stands for Application Programming Interface, lets us write and read twitter data. We can use it for read profile and access our followers data, a high volume of tweets on particular subjects in specific locations.

Imagine, we are at the restaurant on the table, ordering food from the kitchen. Waiter is between kitchen and us who confirmed that kitchen staff gets all orders and meals provided to right customers. We are also free to know about items available before the order. Think about the waiter role, if we don’t have waiter in the restaurants. We need to push our self very hard to the kitchen for the food. That’s where the Application Programming Interface plays their role as a middle man between two diverse software systems or programs.

There are two types of API’s twitter offer as Historical tweets and the other one is Stream Live tweets. The diverse is innate just by their names. The Stream Live, using this tweets API we can takes continuous flow of tweets at the exact moment and record them. The Historical tweets allow us to pick up some portion of the past tweets either a specific user tweets from a certain timeframe.

Using your suggested keywords from part (a) and your knowledge of Twitter, collect a series of Tweets surrounding your chosen brand/famous person and save them to a file. Your collected Tweets should span a minimum ONE-week period.

The first step is to apply for the Twitter developer account and to link to my Twitter account. After getting approval, I need to register my application from the Twitter Developer Website and create an app. I use my application name is W&S_Media_cw.



After that we need to imports the libraries in our coding. I use replit for this project. Following are the four libraries that I am used in this program. “codecs” library allows python program to encode and decode text for diverse representations. “tweepy” library allows python program to utilize the twitter API. “os” and “sys” libraries are helps to determine the paths, directories and filenames.

```
import codecs
import tweepy
import os
import sys
```

This program used for the authentication each request via Twitter API by using specific keys. For the security reason, I hide the keys.

```

#I hidden my API keys for security reason
CONSUMER_KEY = "*****HIDE CONSUMER_KEY*****"
CONSUMER_SECRET = "*****HIDE CONSUMER_SECRET*****"
ACCESS_TOKEN = "*****HIDE ACCESS_TOKEN*****"
ACCESS_TOKEN_SECRET = "*****HIDE ACCESS_TOKEN_SECRET*****"

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
```

The following code will verify CONSUMER_KEY and ACCESS_TOKEN with the Twitter API. When this key valid and works then it will print the command “Authentication OK” but if is there any error occur in validation of this keys then it will print “Error during Authentication”

```
#Make sure that the credentials work
api = tweepy.API(auth)
try:
    api.verify_credentials()
    print("Authentication OK")
except:
    print("Error during authentication")
```

Now, I need to save all the collected tweets using the file name “fibre_tweets.txt”. I save my file with the “utf-8” encoding. I use one of the attributes “a” which is append instead of write “w” so whenever code runs then this file not overwrite but keep on writing from the last line. In some circumstances, there in no any file create by me then it will create one new file making append. Also, I used five keywords to extract the tweets from the tweeter are follows.

```
#to create and append the text in the file
f = codecs.open("fibre_tweets.txt", "a", encoding="utf-8")


# Define the search term and the date_since date as variables
keywords = ['fibre','fiberinternet', 'Internet', 'Broadband', 'Gigabyte']
```

Using the following code, I create a class that defines the tweepy StreamListener and save them in “f” that’s associated the create file name “fibre_tewwts.txt” and the value of the tweets to collected from the Live stream. I collected 1000 tweets per day during six days period using five keywords mentioned in this file.

```python
#tweet listener class
class BasicTwitterListener(tweepy.StreamListener):
    def setup(self):
        self.n_tweet = 0

    def on_status(self, status):
        try:
            cleaned_tweet = status.text.replace("\n", " ")
            if len(cleaned_tweet) < 3:
                return True
            f.write(cleaned_tweet + "\n")
            self.n_tweet += 1
            print(self.n_tweet)

            if self.n_tweet == 1000:
                f.close()
                return False
        except Exception as e:
            print("Exception when reading from stream:")
            pass

    def on_error(self, status_code):
        print("Encountered error with status code: " + str(status_code))
        return True

    def on_timeout(self):
        print("Timed out:")
        return True
```

Using the following code, we must filter the collected tweets using the language only in English and utilized the keywords with the track commend

```python
#assign the class to a variable
l = BasicTwitterListener()
l.setup()

#running the class with filter
live_stream = tweepy.streaming.Stream(auth, l)
live_stream.filter(languages=["en"], follow=None, track=keywords)
```

Running this code one-time each day during six days and 7th day I change code to collect 510 tweets, I will collect a total of 6510 tweets related to the Fibre Optics duration of 7 days.

```
Authentication OK
1
2
3
4
5
6
```

Files

main.py

fibre_tweets.txt

Packager files

poetry.lock

pyproject.toml

fibre_tweets.txt

RT @JayElHarris: It's also extremely weird how the internet treats people existing as "representation" and feels the need to codify existen…

RT @ndekekwe: Elon Musk's SpaceX arrives Nigeria with its Starlink. The team is working with NCC, the industry regulator for permit. Expect…

RT @itsMalikEl: 2007: "don't talk to strangers on the internet!" 2021: "do any mutuals wanna fuck?"

RT @amyklobuchar: In 2021, every family in America should have access to high-speed internet — regardless of their ZIP code.

RT @desithwan: defending faramir on the internet isn't enough, i need boromir

RT @MilaEric1: Bad News about Nigeria & her Government allover the Internet/social media. #BiafraExit .

RT @confusedvichar: New character: Karl Gujjar from Noida. This video series is inspired from a very fascinating YouTube and TikTok sub-g…

RT @Quartzjixler: One of the most detrimental outcomes of the internet and social media is that now people who aren't from Eastern Pennsylv…

RT @StarksGrayson: Ultron is easily the best villain in the mcu, my man spent five minutes on internet and decided the human race needed to…

The coldest video on the internet https://t.co/Y6EfMoymg6

RT @JayElHarris: It's also extremely weird how the internet treats people existing as "representation" and feels the need to codify existen…

RT @cmf_wright: How does your real-life self differ from your internet persona? #WritingCommunity #Writers #WritersCafe #Writing

@existingee @heyGuhRL Wow 2021 lol everyone is so soft now wth it's a joke if being mocked on the internet is too m… https://t.co/q4VvcJTdx7

RT @totalwoke: #MothersDay is just celebration of love. Bhaidooj is celebration of brahminical patriarchy, misogyny, racism, slavery, polar…

RT @itsMalikEl: 2007: "don't talk to strangers on the internet!" 2021: "do any mutuals wanna fuck?"

Using a suitable example, discuss the role of text pre-processing in the context of social media analysis. Identify TWO pre-processing steps relevant to the dataset you created in part (c) and apply them to your dataset.

We need to follow serval step for text pre-processing in the context of social media analysis. [1] filter the language [2] Blank spaces cleaning [3] Lowercase everything [4] Removing the URL [5] Removing the Stop words [6] Removing the Emojis [7] Removing Rt (retweet) [8] Removing the selected punctuations

Now we need to create another program for the pre-processing. First of all, import the libraries. “nltk” is the new libraries we use in this program to specified for updated stop words & language detect

```
import nltk
nltk.download("stopwords")
import codecs
from langdetect import detect
import re
import sys
from nltk.corpus import stopwords

```

After that need to filter languages of the collected tweets using below commends.

```
#checking if the language filter worked properly
def find_eng_tweet(filename):
  with codecs.open(filename, "r", encoding = "utf-8") as f:
    lines = list(f)
    for line in lines[0:]:
      if len(line) > 2:
        try:
          lang = detect(line)
          if lang == "en":
            yield line
        except:
          pass

```

Now the various pre-processing methods are mention in the program with “#”

```
#cleaning the excessive spacing
def clean_t(tweet):
  return tweet.strip()

#lowercase everything
def lowercase_t(tweet):
  return tweet.lower()

```

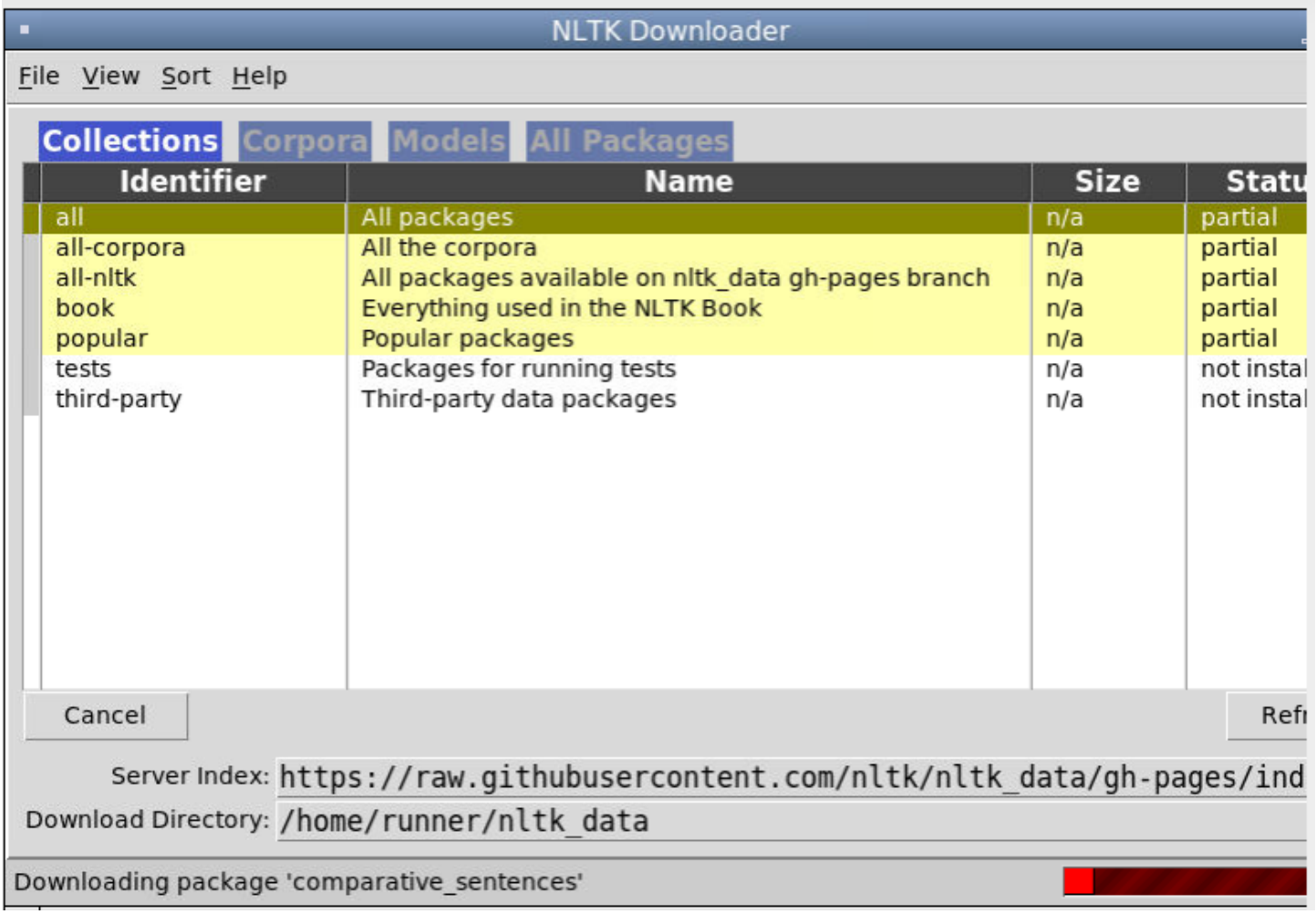
```python
#remove the url
def remove_url(tweet):
  w_keep = []
  for word in tweet.split():
    word = word.strip()
    if word.startswith("http"):
      w_keep.append("")
    else:
      w_keep.append(word)
  return " ".join(w_keep)

#stopwords removal
def remove_stopwords(tweet):
  stop_words = set(stopwords.words('english')) #["is", "a", "an", "in", "on", "am", "are", "and", "i", "the"]
  w_keep = []
  for word in tweet.split():
    word = word.strip()
    if not word in stop_words:
      w_keep.append(word)
  return " ".join(w_keep)

#removing emojis
def remove_emoji(tweet):
  emoji_pattern = re.compile("["
                              u"\U0001F600-\U0001F64F"  # emoticons
                              u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                              u"\U0001F680-\U0001F6FF"  # transport & map symbols
                              u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                              u"\U00002500-\U00002BEF"  # chinese char
                              u"\U00002702-\U000027B0"
                              u"\U00002702-\U000027B0"
                              u"\U000024C2-\U0001F251"
                              u"\U0001f926-\U0001f937"
                              u"\U00010000-\U0010ffff"
                              u"\u2640-\u2642"
                              u"\u2600-\u2B55"
                              u"\u200d"
                              u"\u23cf"
                              u"\u23e9"
                              u"\u231a"
                              u"\ufe0f"  # dingbats
                              u"\u3030"
                              "]+", flags=re.UNICODE)
  return emoji_pattern.sub(r"", tweet)
```

```python
#removing rt
def remove_rt(tweet):
  w_keep = []
  for word in tweet.split():
    word = word.strip()
    if word != "rt":
      w_keep.append(word)
  return " ".join(w_keep)

#removing the punctuations
def remove_punctuation(tweet):
  punctuations = '''!;:'"\,<>()[]{}€$./?%^&*_~’…'''
  no_punct = ""
  for char in tweet:
    if char not in punctuations:
      no_punct = no_punct + char
  return "".join(no_punct)

#calling all the other definition
def processed_tweets(filename):
  for tweet in find_eng_tweet(filename):
    tweet = clean_t(tweet)
    tweet = lowercase_t(tweet)
    tweet = remove_url(tweet)
    tweet = clean_t(tweet)
    tweet = remove_stopwords(tweet)
    tweet = remove_emoji(tweet)
    tweet = remove_rt(tweet)
    tweet = remove_punctuation(tweet)
    yield tweet

file_in = "fibre_tweets.txt"
file_out = "fibretweets_processed.txt"

#exporting everything on another file
with codecs.open(file_out, "w", encoding = "utf-8") as f:
  for tweet in processed_tweets(file_in):
    f.write(tweet + "\n")

print("Process Complete")

stop_words = set(stopwords.words('english'))
print(stop_words)
```

For running this program, we need to install some components from the nltk using the shell commend prompt from the replit so that we can success and upgrade the pip

```
~/w1813148CWP02$ pip install numpy
Requirement already satisfied: numpy in /opt/virtualenvs/python3/lib/python3.8/site-packages
(1.20.2)
WARNING: You are using pip version 19.3.1; however, version 21.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
~/w1813148CWP02$ pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/cd/6f/43037c7bcc8bd8ba7c9074256b1a11596
daa15555808ec748048c1507f08/pip-21.1.1-py3-none-any.whl (1.5MB)
     |                                        | 1.6MB 3.4MB/s
ERROR: dephell 0.8.3 has requirement pip<=19.3.1,>=18.0, but you'll have pip 21.1.1 which is
incompatible.
Installing collected packages: pip
  Found existing installation: pip 19.3.1
    Uninstalling pip-19.3.1:
      Successfully uninstalled pip-19.3.1
Successfully installed pip-21.1.1
~/w1813148CWP02$ pip install numpy
Requirement already satisfied: numpy in /opt/virtualenvs/python3/lib/python3.8/site-packages
(1.20.2)
~/w1813148CWP02$ pip install nltk
Requirement already satisfied: nltk in /opt/virtualenvs/python3/lib/python3.8/site-packages (
3.6.2)
Requirement already satisfied: click in /opt/virtualenvs/python3/lib/python3.8/site-packages
(from nltk) (7.1.2)
Requirement already satisfied: regex in /opt/virtualenvs/python3/lib/python3.8/site-packages
(from nltk) (2021.4.4)
Requirement already satisfied: tqdm in /opt/virtualenvs/python3/lib/python3.8/site-packages (
from nltk) (4.60.0)
Requirement already satisfied: joblib in /opt/virtualenvs/python3/lib/python3.8/site-packages
 (from nltk) (1.0.1)
~/w1813148CWP02$ python
Python 3.8.9 (default, May  3 2021, 02:40:41)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download()
~/w1813148CWP02$
```

When we run this above commends then below prompt come up with the NLTK Downloader so we press the Download button after that downloading starts. This is very important for run this code.



We got processed tweets. Processed tweets would be neat and clean

fibretweets_processed.txt

```
@balloonwanted osen shares gwsn making comeback may 26th gwsn making comeback complete 7
sothe tories lie steal kill people hartlepool still vote them party taken
could french vaccine thats said variant proof live hype
wais chief executive step aside gymnastics probe historical allegations mistreatment
lion - #responsive bootstrap 4 #admin dashboard template light amp dark sidebar version horizontal nav version
@blakes7bot #blakes7 series c episode 07 - children auron ca-2 storage bio replication plant cally zelda th
@msjessienz comes broadband plan never watch it might consider switching soon
@wildfirstcanada bc open net pen fish farms spread deadly diseases threaten wild pacific salmon end pens support clean s
@balloonwanted #update fnc entertainment confirm sf9 members staff tested negative covid-19 following preemptiv
@131label inquiries related bis activity supporting please send email via official mail address  gtid@131labeln
@131label inquiries related bis activity supporting please send email via official mail address  gtid@131labeln
```

```
[nltk_data] Downloading package stopwords to /home/runner/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Process Complete
```

Create a Python program to count the most commonly used words in your dataset and use it to generate a "word cloud".

```python
import codecs
import re
from collections import Counter
import os
import sys

#importing the processed tweets
f = codecs.open("fibretweets_processed.txt", "r", encoding="utf-8")
data = f.read()
f.close()


#word counting definition
def word_count(str):
    counts = Counter()
    words = str.split()
    for word in words:
        if (word in counts
            ) and not (word.startswith("@")) and not (word.startswith("#")):
            counts[word] += 1
        else:
            counts[word] = 1
    return counts


#tag counting definition
def tag_count(str):
    counts = Counter()
    words = str.split()
    for word in words:
        if (word in counts) and (word.startswith("#")):
            counts[word] += 1
        else:
            counts[word] = 1
    return counts

#most mentioned user counting
def user_count(str):
    counts = Counter()
    words = str.split()
    for word in words:
        if (word in counts) and (word.startswith("@")):
            counts[word] += 1
        else:
            counts[word] = 1
    return counts

```

```python
counter = word_count(data)
tags = tag_count(data)
users = user_count(data)
print("\n", counter.most_common(20), "\n\n", tags.most_common(20), "\n\n",
        users.most_common(20), "\n\n")

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

#preparation for the plot
word, w_counts = zip(*counter.most_common(10))
tag, t_counts = zip(*tags.most_common(10))
user, u_counts = zip(*users.most_common(10))

#importing the libraries to plot the table
import pandas as pd
import matplotlib.pyplot as plt

#creating dataframe for each counting
wordDF = pd.DataFrame({"word": word, "w_count": w_counts})
tagDF = pd.DataFrame({"tags": tag, "t_count": t_counts})
userDF = pd.DataFrame({"users": user, "u_count": u_counts})

#plotting tag counting
ax = tagDF.plot(x="tags",
                y=["t_count"],
                kind="bar",
                figsize=(20, 25),
                fontsize=12)
ax.set_xlabel("Tag")
ax.set_ylabel("Frequency")
plt.savefig("tag_count.png")

#plotting word counting
ax = wordDF.plot(x="word",
                 y=["w_count"],
                 kind="bar",
                 figsize=(20, 15),
                 fontsize=12)
ax.set_xlabel("Word")
ax.set_ylabel("Frequency")
plt.savefig("word_count.png")
```

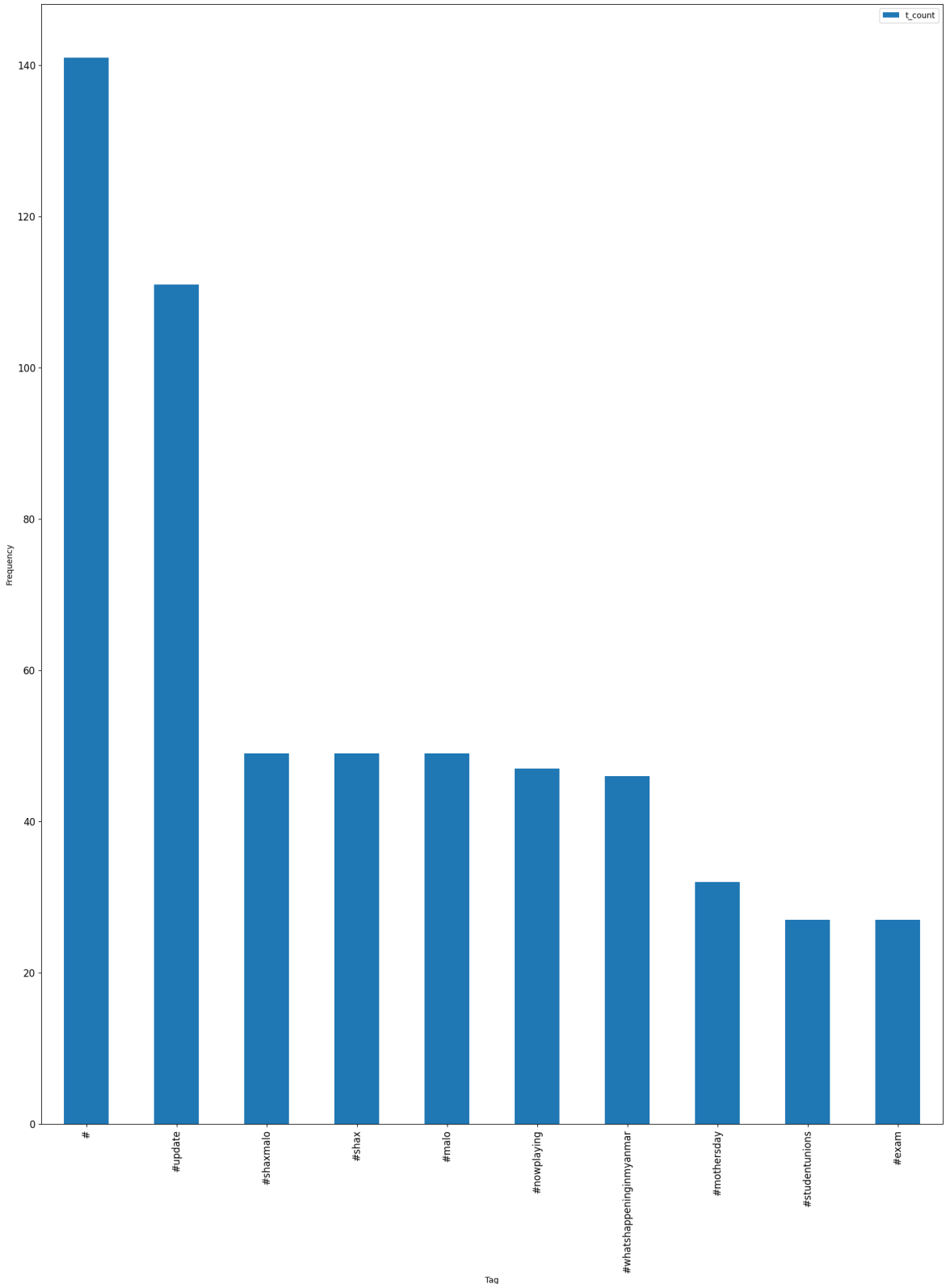
```python
#plotting user counting
ax = userDF.plot(x="users",
                 y=["u_count"],
                 kind="bar",
                 figsize=(20, 15),
                 fontsize=12)
ax.set_xlabel("User")
ax.set_ylabel("Frequency")
plt.savefig("user_count.png")


#Listing all the tags
def all_tags(str):
    hashtags = []
    words = str.split()
    for word in words:
        if word.startswith("#"):
            hashtags.append(word)
        else:
            pass
    return " ".join(hashtags)


hash_tags = all_tags(data)
print(hash_tags.upper())
```

Word count , tag count and user count Out put

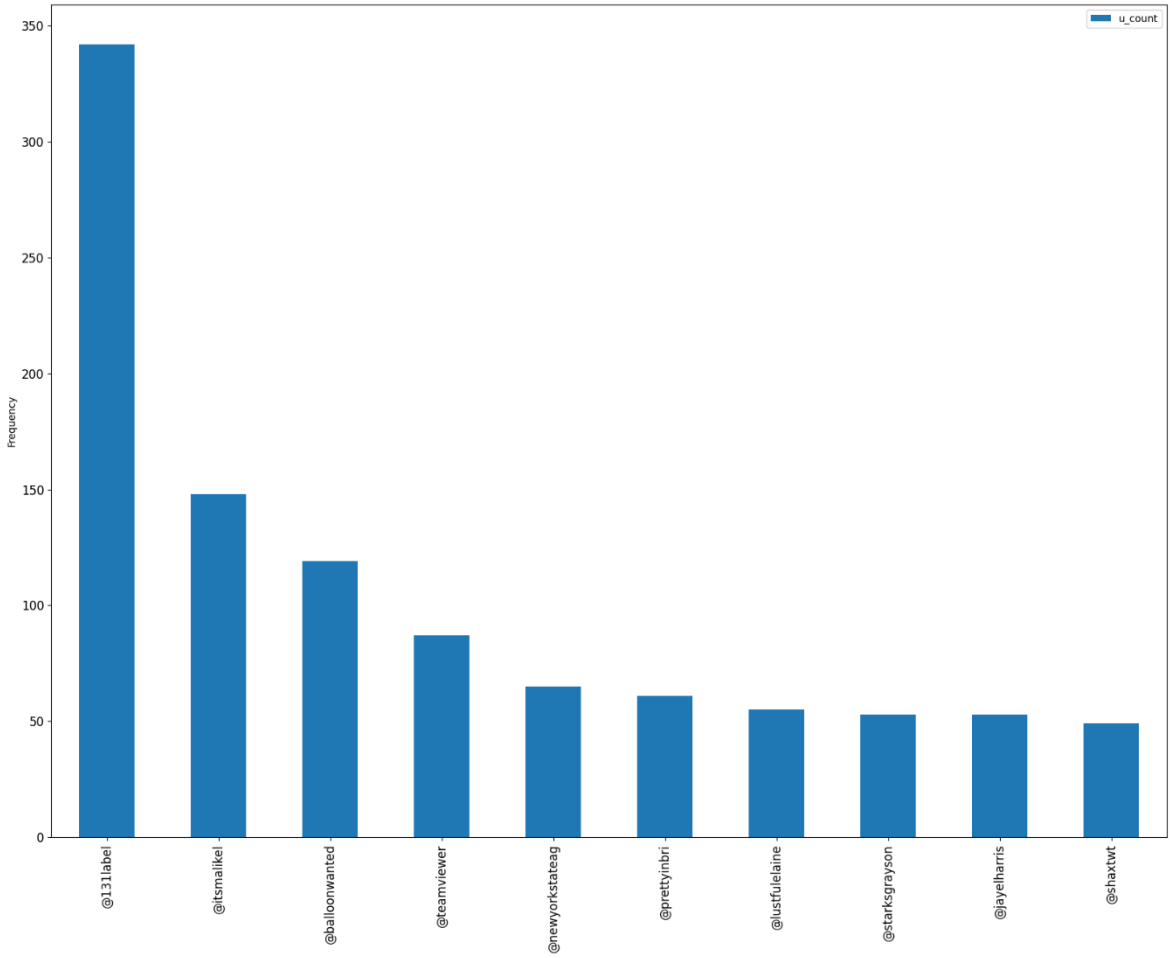
```
 [('internet', 1198), ('-', 577), ('net', 434), ('via', 431), ('please', 413), ('official', 3
55), ('send', 351), ('activity', 348), ('related', 346), ('email', 345), ('mail', 344), ('add
ress', 343), ('inquiries', 340), ('bis', 340), ('supporting', 340), ('gtid@131labeln', 340),
('people', 269), ('amp', 269), ('2021', 258), ('wanna', 225)]

 [('#', 141), ('#update', 111), ('#shaxmalo', 49), ('#shax', 49), ('#malo', 49), ('#nowplayin
g', 47), ('#whatshappeninginmyanmar', 46), ('#mothersday', 32), ('#studentunions', 27), ('#ex
am', 27), ('#astro', 26), ('#mdcat', 26), ('#dogecoin', 25), ('#helixstudios', 23), ('#mellow
975xmss', 21), ('#affiliatemarketing', 17), ('#cardano', 16), ('#radio', 16), ('#cybersecurit
y', 16), ('#job', 15)]

 [('@131label', 342), ('@itsmalikel', 148), ('@balloonwanted', 119), ('@teamviewer', 87), ('@
newyorkstateag', 65), ('@prettyinbri', 61), ('@lustfulelaine', 55), ('@starksgrayson', 53), (
'@jayelharris', 53), ('@shaxtwt', 49), ('@teamviewerhelp', 47), ('@piichi02', 40), ('@rellrom
ance', 39), ('@ksgupdates', 28), ('@mattdpearce', 27), ('@haqooqekhalq', 27), ('@mjibrannasir
', 26), ('@astrofancafe', 23), ('@totalwoke', 23), ('@', 20)]
```

I also get graphs as a outputs of the program

Tag count


t_count
Frequency
0
20
40
60
80
100
120
140
#
#update
#shaxmalo
#shax
#malo
#nowplaying
#whatshappeninginmyanmar
#mothersday
#studentunions
#exam
Tag

User Count



Total word count after pre-processing

```
6256
```

Word Cloud using the word count words via worditout.com

Use your processed data file to produce a series of graphs or charts to summaries the following information.

- ***I. The number of tweets posted per day***
- ***II. The number of unique users per day***
- ***III. The top 10 most active users over the entire period***

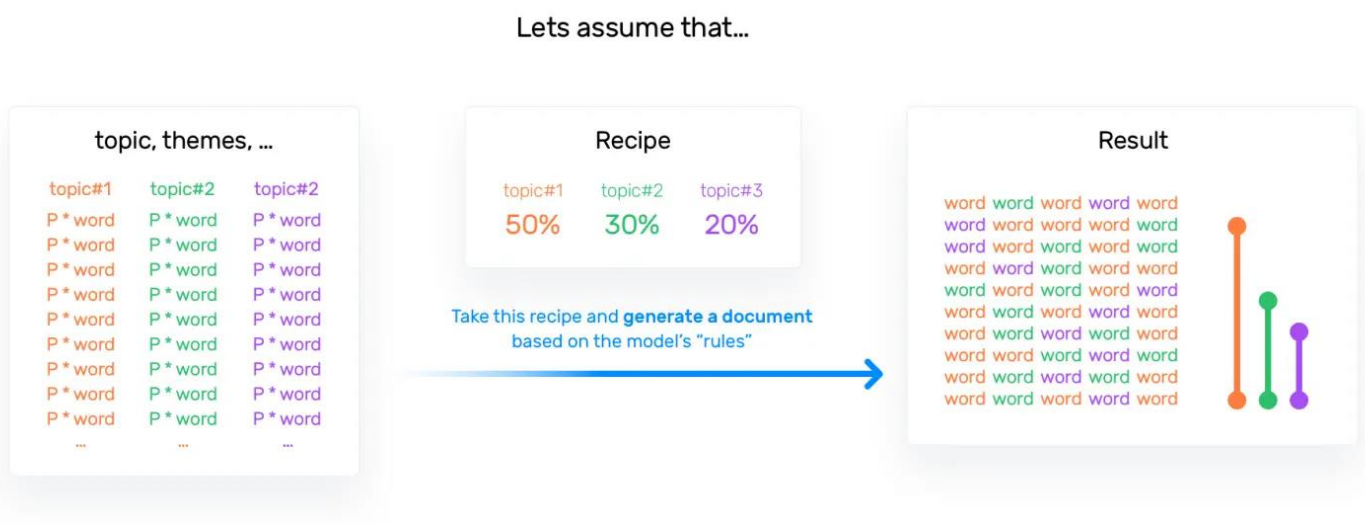
Using a suitable approach, construct a LDA topic model to identify themes of discussion within your dataset.

Topic modeling is a one type of machine learning technique that analyze text data automatically using cluster words for set of documents. This is unsupervised machine learning. It's a easy and quick way to analyze our data. Counting words and similar word grouping are part of the topic modeling within unstructured data.

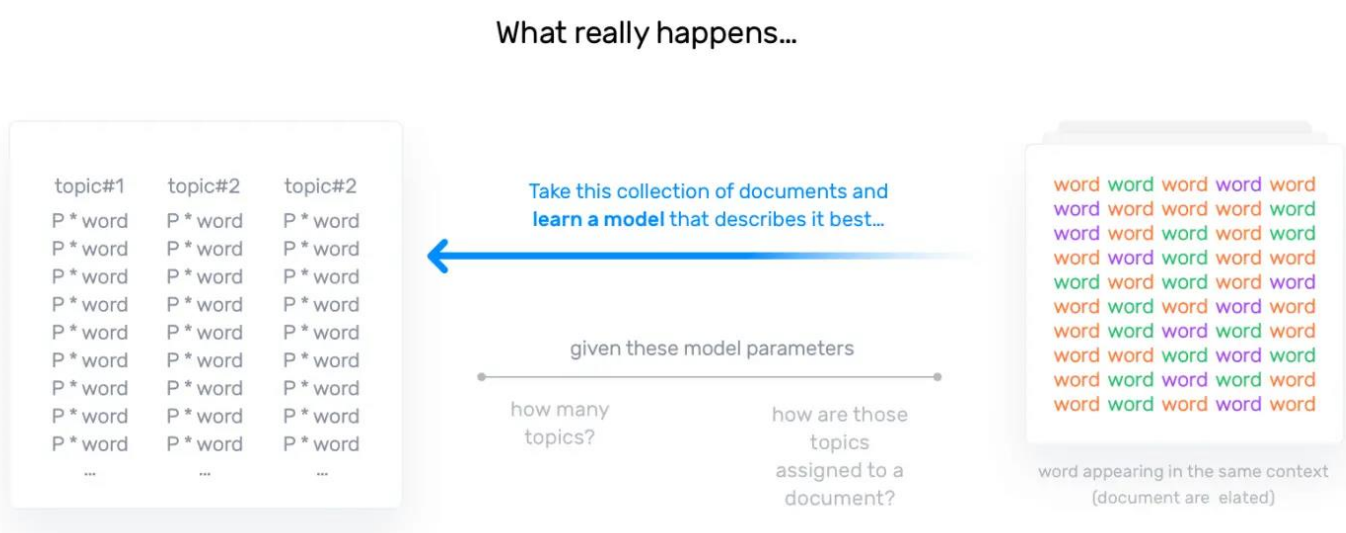
Topic modeling refers to the process of dividing a corpus of documents in two:[3]

1. A list of the topics covered by documents in the corpus
2. Serval set of documents from the corpus by topics they cover.

LDA assumes that topics and documents look like this:[3]



And, when LDA models a new document, it works this way:[3]



We, used the tweet cleaning process once but still in this step we can cleaning some of the data cleaning using R. first of all, I installed the missing packages that required to run the program

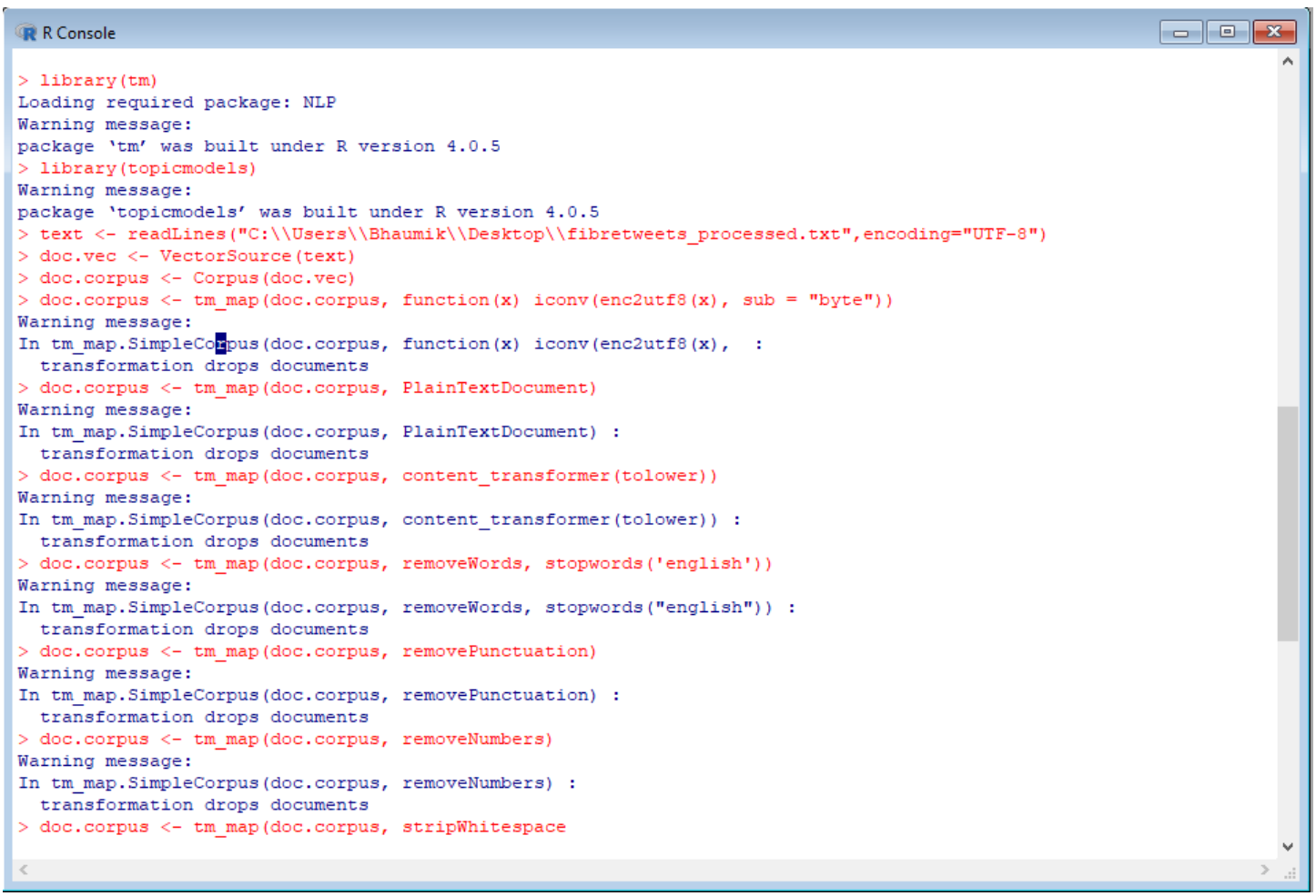
Install.packages(c(“tm”))

Install.packages(c(“topicmodels”))

text <- readLines("C:\\Users\\Bhaumik\\Desktop\\fibretweets_processed.txt ",encoding="UTF-8")

doc.vec <- VectorSource(text)
doc.corpus <- Corpus(doc.vec)
doc.corpus <- tm_map(doc.corpus, function(x) iconv(enc2utf8(x), sub = "byte"))
doc.corpus <- tm_map(doc.corpus, PlainTextDocument)
doc.corpus <- tm_map(doc.corpus, content_transformer(tolower))
doc.corpus <- tm_map(doc.corpus, removeWords, stopwords('english'))
doc.corpus <- tm_map(doc.corpus, removePunctuation)
doc.corpus <- tm_map(doc.corpus, removeNumbers)
doc.corpus <- tm_map(doc.corpus, stripWhitespace)

This is the outputs from the R were running the above code



```
R Console
> library(tm)
Loading required package: NLP
Warning message:
package 'tm' was built under R version 4.0.5
> library(topicmodels)
Warning message:
package 'topicmodels' was built under R version 4.0.5
> text <- readLines("C:\\Users\\Bhaumik\\Desktop\\fibretweets_processed.txt",encoding="UTF-8")
> doc.vec <- VectorSource(text)
> doc.corpus <- Corpus(doc.vec)
> doc.corpus <- tm_map(doc.corpus, function(x) iconv(enc2utf8(x), sub = "byte"))
Warning message:
In tm_map.SimpleCorpus(doc.corpus, function(x) iconv(enc2utf8(x),  :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, PlainTextDocument)
Warning message:
In tm_map.SimpleCorpus(doc.corpus, PlainTextDocument) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, content_transformer(tolower))
Warning message:
In tm_map.SimpleCorpus(doc.corpus, content_transformer(tolower)) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, removeWords, stopwords('english'))
Warning message:
In tm_map.SimpleCorpus(doc.corpus, removeWords, stopwords("english")) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(doc.corpus, removePunctuation) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(doc.corpus, removeNumbers) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, stripWhitespace
```

```
R Console
> doc.corpus <- tm_map(doc.corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(doc.corpus, removePunctuation) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(doc.corpus, removeNumbers) :
  transformation drops documents
> doc.corpus <- tm_map(doc.corpus, stripWhitespace
+ doc.corpus <- tm_map(doc.corpus, stripWhitespace)
Error: unexpected symbol in:
"doc.corpus <- tm_map(doc.corpus, stripWhitespace
doc.corpus"
> doc.corpus <- tm_map(doc.corpus, stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(doc.corpus, stripWhitespace) :
  transformation drops documents
> dtm <- DocumentTermMatrix(doc.corpus)
> dtm <- removeSparseTerms(dtm, 0.98)
> x <- as.matrix(dtm)
> x <- x[which(rowSums(x) > 0),]
> rownames(x) <- 1:nrow(x)
> lda <- LDA(x, <<nooftopics>>)
Error: unexpected input in "lda <- LDA(x, <<"
> lda <- LDA(x,6)
> terms(lda, 6)
     Topic 1 Topic 2     Topic 3      Topic 4    Topic 5  Topic 6
[1,] "one"   "internet"  "wanna"      "official" "net"    "via"
[2,] "like"  "amp"       """          "send"     "people" "please"
[3,] "time"  "new"       "talk"       "label"    "update" "broadband"
[4,] "need"  "broadband" "strangers"  "activity" "covid"  "new"
[5,] "get"   "best"      "internet”"  "related"  "day"    "internet"
[6,] "may"   "dont"      "itsmalikel" "email"    "live"   "email"
> perplexity(lda)
[1] 22.69929
```

From this analysis, we got the most talked 6 topics and perplexity LDA value 22.69929

Apply noun phrase recognition to your dataset and identify the top five most mentioned noun phrases. Construct a sentiment model for each of your identified noun phrases and compare and contrast the differences in both polarity and sentiment

```python
from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer
import codecs
from textblob.np_extractors import ConllExtractor
import nltk
nltk.download('brown')
nltk.download('movie_reviews')
nltk.download('punkt')

prebuilt_classifier = NaiveBayesAnalyzer()
extractor = ConllExtractor()

f = codecs.open("fibretweets_processed.txt", encoding = "utf-8")

positive = 0
negative = 0
neutral = 0
np = {}

#defining the terms to use
for line in f.readlines():
  lc_line = line.lower()

  if "internet" in lc_line and "2021" in lc_line:
    tweet = TextBlob(lc_line, analyzer = prebuilt_classifier)

    for n in tweet.noun_phrases:
      if n in np:
        np[n] += 1
      else:
        np[n] = 1

    if tweet.sentiment.p_pos >= 0.7:
      positive += 1
    elif tweet.sentiment.p_neg >= 0.7:
      negative += 1
    else:
      neutral += 1

#sorting the noun phrases by the highest appearance
for w in sorted(np, key = np.get, reverse = True) [0:5]:
  print(w, np[w])
```

```python
#print out the values
print("\n", positive, negative, neutral)
total = positive + negative + neutral
positivePercent = positive/total*100
negativePercent = negative/total*100
print("Percent of positive Tweets: ", round(positivePercent, 2), "%")
print("Percent of negative Tweets: ", round(negativePercent, 2), "%")
```

Outputs of the program

```
[nltk_data] Downloading package brown to /home/runner/nltk_data...
[nltk_data]   Package brown is already up-to-date!
[nltk_data] Downloading package movie_reviews to
[nltk_data]     /home/runner/nltk_data...
[nltk_data]   Package movie_reviews is already up-to-date!
[nltk_data] Downloading package punkt to /home/runner/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
@ itsmalikel 147
" dont talk strangers internet " 147
mutuals wan na fuck " 147
@ amyklobuchar 19
family america access 19

 29 149 3
Percent of positive Tweets:  16.02 %
Percent of negative Tweets:  82.32 %
>
```

During the coursework,

References

[1] Technical University of Munich, Date 21 Feb 2019
https://www.technologist.eu/new-record-data-transfer-speed-in-fiber-optic-network/#:~:text=In%20an%20intercity%20field%20experiment,s)%20with%20a%20single%20wavelength.

[2] Figure 1 and 2: Mark Jackson from Dorset (England)
https://www.ispreview.co.uk/index.php/2020/09/uk-trails-as-10-countries-pass-95-full-fibre-broadband-cover.html

[3] Introduction to Topic Modeling, Federico Pascual, 26th September 2019
https://monkeylearn.com/blog/introduction-to-topic-modeling/#:~:text=Topic%20modeling%20is%20an%20unsupervised,characterize%20a%20set%20of%20documents.