Twitter Data Analysis ¶

In this project, we will analyse the tweets with the most trending hashtag in New Delhi.

So there are 3 steps involved in this process.

1. Getting the data from Twitter
2. Present the data in a proper format so that it can be analysed easily
3. Analyse the data

We will be working with python and Twitter API.

Extracting data from twitter

I have used Twitter API tweepy for doing this. So, we will start with installing tweepy.

In [3]:
```
pip install tweepy
```

```
Requirement already satisfied: tweepy in c:\programdata\anaconda3\lib\site-pack
ages (3.10.0)
Requirement already satisfied: requests[socks]>=2.11.1 in c:\programdata\anacon
da3\lib\site-packages (from tweepy) (2.24.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: six>=1.10.0 in c:\programdata\anaconda3\lib\site
-packages (from tweepy) (1.15.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\programdata\anaco
nda3\lib\site-packages (from tweepy) (1.3.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\pr
ogramdata\anaconda3\lib\site-packages (from requests[socks]>=2.11.1->tweepy)
(1.25.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\l
ib\site-packages (from requests[socks]>=2.11.1->tweepy) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in c:\programdata\anaconda3\lib\sit
e-packages (from requests[socks]>=2.11.1->tweepy) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\programdata\anaconda3\li
b\site-packages (from requests[socks]>=2.11.1->tweepy) (3.0.4)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6; extra == "socks" in c:\p
rogramdata\anaconda3\lib\site-packages (from requests[socks]>=2.11.1->tweepy)
(1.7.1)
Requirement already satisfied: oauthlib>=3.0.0 in c:\programdata\anaconda3\lib
\site-packages (from requests-oauthlib>=0.7.0->tweepy) (3.1.0)
```

Now, make a developer account on twitter. You might need to wait for some time before it gets approved

In the developer account click on "Create a new App" and you will be given credentials.

Now , we will include those credentials to make GET requests from twitter api.

In [4]:
```
import tweepy
consumer_key= 'YBA2m8JJ9cy3E9wXxVi2yYcX8'
consumer_secret= 'HFMeoWIOhDZAjbDpfgAyujxSSZEQI5BAOuGxPiQRT4HniPk9cq'
access_token= '1342378688453648384-tVtbtEATUh3NpZWttbaoKFp438NXpI'
access_token_secret= 'FTvj3KU0oJCe9ZJBGGnKDAMPhU6eICcqV7GqfUGu1a68Y'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api=tweepy.API(auth)
```

Now, to avoid running our program repetitively, we will want to store all the tweets in a file. So we will create a new text file "json_dumps.txt"

In [5]:
```
import os
import json
import sys
import tweepy
import io

File = io.open('json_dumps.txt', 'w', encoding="utf-8")
```

Now we will make the GET request. Note three things:

1. We are using the hashtag "cricket" as, without retweets we can get more than 10000 tweets.
2. We will be using a filter for retweets(for uniqueness)

In [6]:
```
tweet_data =  tweepy.Cursor(api.search,q="#cricket"+" -filter:retweets",geocode='
#tweet_data does extract the top trending hashtag win a particular location and w
#(> 10000). So for now I have considered a really popular topic and location to b


tweet_data =  tweepy.Cursor(api.search,q="#cricket"+" -filter:retweets", count=11
```

We will be storing the tweets in a csv file as well. For this we will use pandas.So , using open() we create a new csv file and then import pandas and create a dataframe.

In [7]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-pack
ages (1.1.3)
Requirement already satisfied: pytz>=2017.2 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas) (2020.1)
Requirement already satisfied: numpy>=1.15.4 in c:\programdata\anaconda3\lib\si
te-packages (from pandas) (1.19.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\programdata\anacond
a3\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7.3->pandas) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [8]:
```
import pandas as pd
df = pd.DataFrame()

csv_file = io.open("tweets_info.csv", "w", encoding="utf-8")
```

Now , we will iterate through all the tweets in the tweet_data object. Every tweet has a *json* attribute to it that contains all the info about the tweet. So we will write the json strings to the text file and the csv file.

In [9]:
```
i = 0;
for tweet in  tweet_data:
    if (hasattr(tweet , "_json")):
        i += 1;
        File.write(json.dumps(tweet._json))
        File.write("\n")
        df_2 = pd.json_normalize(tweet._json)
        df = df.append(df_2, ignore_index=True)
        print(i)
        if (i >=11000):
            print(i)
            break;
export_csv = df.to_csv(csv_file, index = None, header=True)
```

```
10486
10487
10488
10489
10490
10491
10492
10493
10494
10495
10496
10497
10498
10499
10500
10501
10502
10503
10504
10505
```

In [10]:
```
print(i)
```

```
11000
```

Voila!! We have got all the tweets and we have got them in a suitable format too

Analysis of the data.

Now we have got all the stuff in csv file. So let's start with making a wordcloud for all the tweets. Start with getting text of all the posts in the tweets. For this, to pevent the need of repititive computation, we will first store all the words of the posts in a text file.

My initial approach was iterating through each tweet, extracting it's post text as a string and then split the string with spaces to get tokens. But the process is too slow and time taking. So for this, we will use spacy module in python.

So start with installing it through pip on your system.

In [11]:
```
pip install -U spacy
```

```
Requirement already up-to-date: spacy in c:\programdata\anaconda3\lib\site-pack
ages (2.3.5)
Requirement already satisfied, skipping upgrade: murmurhash<1.1.0,>=0.28.0 in
c:\programdata\anaconda3\lib\site-packages (from spacy) (1.0.5)
Requirement already satisfied, skipping upgrade: numpy>=1.15.0 in c:\programdat
a\anaconda3\lib\site-packages (from spacy) (1.19.2)
Requirement already satisfied, skipping upgrade: blis<0.8.0,>=0.4.0 in c:\progr
amdata\anaconda3\lib\site-packages (from spacy) (0.7.4)
Requirement already satisfied, skipping upgrade: srsly<1.1.0,>=1.0.2 in c:\prog
ramdata\anaconda3\lib\site-packages (from spacy) (1.0.5)
Requirement already satisfied, skipping upgrade: preshed<3.1.0,>=3.0.2 in c:\pr
ogramdata\anaconda3\lib\site-packages (from spacy) (3.0.5)
Requirement already satisfied, skipping upgrade: catalogue<1.1.0,>=0.0.7 in
c:\programdata\anaconda3\lib\site-packages (from spacy) (1.0.0)
Requirement already satisfied, skipping upgrade: tqdm<5.0.0,>=4.38.0 in c:\prog
ramdata\anaconda3\lib\site-packages (from spacy) (4.50.2)
Requirement already satisfied, skipping upgrade: plac<1.2.0,>=0.9.6 in c:\progr
amdata\anaconda3\lib\site-packages (from spacy) (1.1.3)
Requirement already satisfied, skipping upgrade: wasabi<1.1.0,>=0.4.0 in c:\pro
gramdata\anaconda3\lib\site-packages (from spacy) (0.8.0)
Requirement already satisfied, skipping upgrade: requests<3.0.0,>=2.13.0 in
c:\programdata\anaconda3\lib\site-packages (from spacy) (2.24.0)
Requirement already satisfied, skipping upgrade: cymem<2.1.0,>=2.0.2 in c:\prog
ramdata\anaconda3\lib\site-packages (from spacy) (2.0.5)
Requirement already satisfied, skipping upgrade: thinc<7.5.0,>=7.4.1 in c:\prog
ramdata\anaconda3\lib\site-packages (from spacy) (7.4.5)
Requirement already satisfied, skipping upgrade: setuptools in c:\programdata\a
naconda3\lib\site-packages (from spacy) (50.3.1.post20201107)
Requirement already satisfied, skipping upgrade: urllib3!=1.25.0,!=1.25.1,<1.2
6,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0,>
=2.13.0->spacy) (1.25.11)
Requirement already satisfied, skipping upgrade: certifi>=2017.4.17 in c:\progr
amdata\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2020.
6.20)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in c:\programdata
\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.10)
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in c:\progra
mdata\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

In [12]:
```
!pip install https://github.com/explosion/spacy-models/releases/download/en_core_
```

```
Collecting https://github.com/explosion/spacy-models/releases/download/en_cor
e_web_sm-2.2.0/en_core_web_sm-2.2.0.tar.gz (https://github.com/explosion/spac
y-models/releases/download/en_core_web_sm-2.2.0/en_core_web_sm-2.2.0.tar.gz)
  Downloading https://github.com/explosion/spacy-models/releases/download/en_
core_web_sm-2.2.0/en_core_web_sm-2.2.0.tar.gz (https://github.com/explosion/s
pacy-models/releases/download/en_core_web_sm-2.2.0/en_core_web_sm-2.2.0.tar.g
z) (12.0 MB)
Requirement already satisfied (use --upgrade to upgrade): en-core-web-sm==2.
2.0 from https://github.com/explosion/spacy-models/releases/download/en_core_
web_sm-2.2.0/en_core_web_sm-2.2.0.tar.gz (https://github.com/explosion/spacy-
models/releases/download/en_core_web_sm-2.2.0/en_core_web_sm-2.2.0.tar.gz) in
c:\programdata\anaconda3\lib\site-packages
Requirement already satisfied: spacy>=2.2.0 in c:\programdata\anaconda3\lib\s
ite-packages (from en-core-web-sm==2.2.0) (2.3.5)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in c:\programdata\anacond
a3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (0.8.0)
Requirement already satisfied: thinc<7.5.0,>=7.4.1 in c:\programdata\anaconda
3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (7.4.5)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in c:\programdata\anac
onda3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (1.0.0)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in c:\programdata\anaconda3
\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (1.1.3)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\programdata\an
aconda3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (1.0.5)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\programdata\anaconda
3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (2.0.5)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in c:\programdata\anaconda3
\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (0.7.4)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\programdata\anac
onda3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (2.24.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\programdata\anaconda
3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (4.50.2)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\sit
e-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (50.3.1.post20201107)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\programdata\anacon
da3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (3.0.5)
Requirement already satisfied: numpy>=1.15.0 in c:\programdata\anaconda3\lib
\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (1.19.2)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in c:\programdata\anaconda
3\lib\site-packages (from spacy>=2.2.0->en-core-web-sm==2.2.0) (1.0.5)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3
\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.0->en-core-web-s
m==2.2.0) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
 c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->sp
acy>=2.2.0->en-core-web-sm==2.2.0) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in c:\programdata\anaconda3\lib\s
ite-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.0->en-core-web-sm==2.
2.0) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\programdata\anaconda3
\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.0->en-core-web-s
m==2.2.0) (3.0.4)
Building wheels for collected packages: en-core-web-sm
  Building wheel for en-core-web-sm (setup.py): started
```

```
  Building wheel for en-core-web-sm (setup.py): finished with status 'done'
  Created wheel for en-core-web-sm: filename=en_core_web_sm-2.2.0-py3-none-an
y.whl size=12019126 sha256=d9c2fd094eee92a80a5144223f7bae0f086232d5ed385e5573
79ac70716e3b6b
  Stored in directory: c:\users\ayush\appdata\local\pip\cache\wheels\fc\31\e9
\092e6f05b2817c9cb45804a3d1bf2b9bf6575742c01819337c
Successfully built en-core-web-sm
```

Import all required modules first. And then read the info stored in the tweets.text column in the csv file into a dataframe.

In [13]:
```
import os
import spacy
import io
from spacy.lang.en import English
# Use it on Jupyter Notebook or Google Colab
# DIR_PATH = os.getcwd()
# Use it on Python module
DIR_PATH = os.path.dirname("__file__")

FILE_PATH = r"C:\Users\ayush\OneDrive\Desktop\PRECog\Q2\tweets_info.csv"

import pandas as pd

# Read the file
df = pd.read_csv("tweets_info.csv")
```

Now iterate through all rows in the dataframe and save all the individual words in a list.

In [14]:
```
import en_core_web_sm


final_list = []
for i in range(11000):
    try:
        first_dialogue = df.loc[i, "text"]
        if first_dialogue == None:
            break;
        # use spacy with the  parse
        nlp = en_core_web_sm.load()
        [str(sent) for sent in nlp(first_dialogue).sents]


        # use spacy with the sentencizer
        nlp = English()  # just the language with no model
        sentencizer = nlp.create_pipe("sentencizer")
        nlp.add_pipe(sentencizer)
        k = [str(sent) for sent in nlp(first_dialogue).sents]
        for K in k:
            final_list.append(K)
            print(i)
    except:
        continue;
```

```
10610
10611
10611
10612
10612
10612
10612
10613
10613
10613
10614
10615
10616
10617
10617
10618
10619
10619
10620
10620
```

Now save this list in a text file for future analysis.

In [15]:
```
with io.open("list_of_posts.txt", "w", encoding="utf-8") as f:
    f.write((' '.join([str(elem) for elem in final_list])))
```

Now, we can start with making a word cloud for all the posts related to the hashtag cricket.

Start with importing all the files.

In [16]:
```
pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\programdata\anaconda3\lib\site-p
ackages (1.8.1)Note: you may need to restart the kernel to use updated package
s.
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-
packages (from wordcloud) (3.3.2)
Requirement already satisfied: numpy>=1.6.1 in c:\programdata\anaconda3\lib\sit
e-packages (from wordcloud) (1.19.2)

Requirement already satisfied: pillow in c:\programdata\anaconda3\lib\site-pack
ages (from wordcloud) (8.0.1)
Requirement already satisfied: certifi>=2020.06.20 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (2020.6.20)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib->wordcloud) (1.3.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\p
rogramdata\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-package
s (from cycler>=0.10->matplotlib->wordcloud) (1.15.0)
```

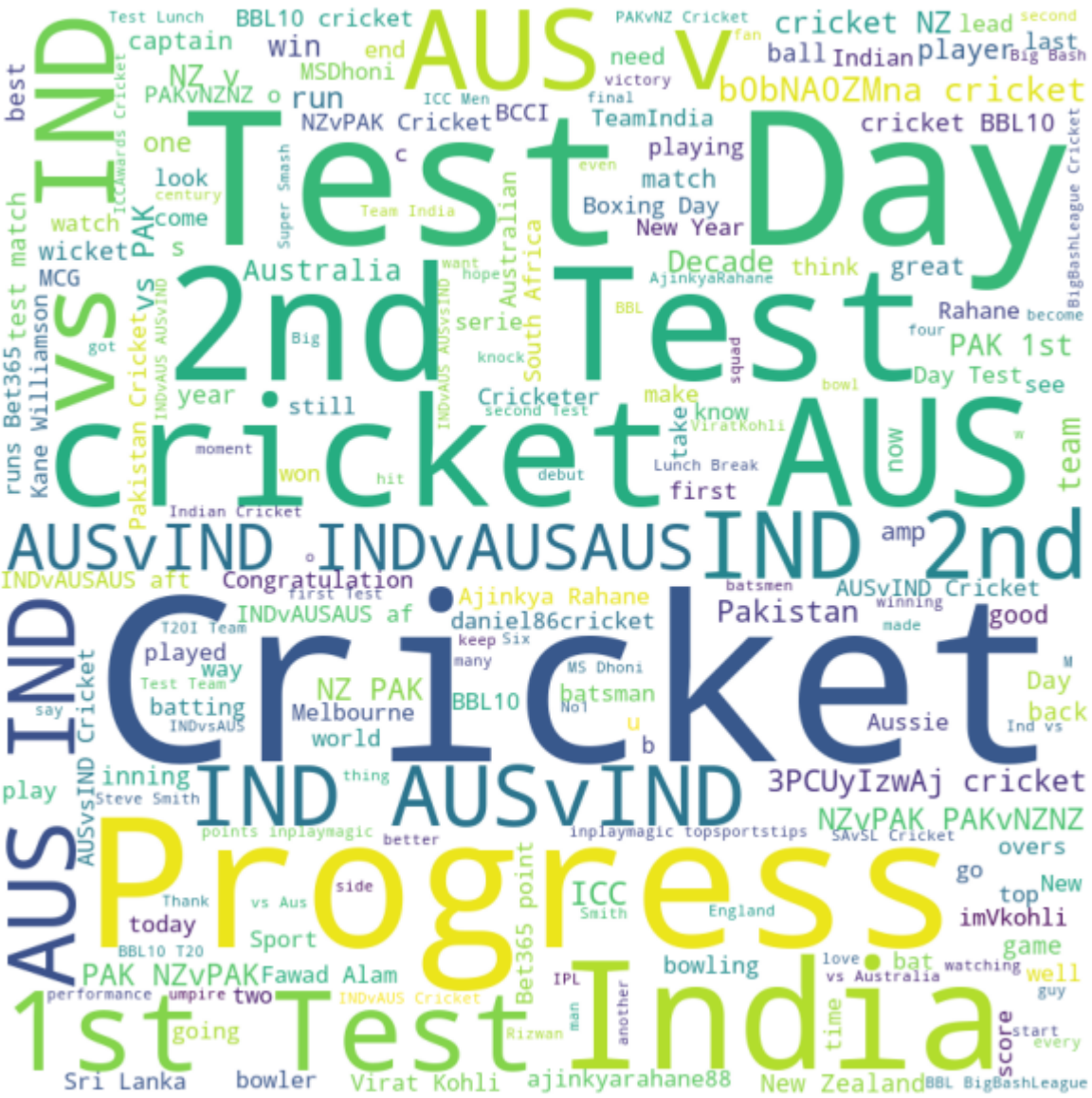
In [17]:
```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd
import io


```

Store the data of the posts in a string.

In [18]:
```
with io.open("list_of_posts.txt", "r", encoding="utf-8") as file:
    data = file.read().replace('\n', '')
```

Now use the wordcloud and pyplot module to display the wordCloud.

In [19]:
```
from wordcloud import WordCloud, STOPWORDS
stopwords = set(STOPWORDS)
wordcloud = WordCloud(width = 800, height = 800, background_color ='white', stopw
stopwords.add("https")
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Now, lets analyse make a word cloud for the description of the users who tweeted related to cricket. We will plot histograms for the same.

So we will repeat the same process as before.

So we can infer the follwing from the above wordCloud.

1. AUS vs IND match is trending.
2. Ajyinke Rahane is trenfing
3. Pakistan, New Zealand, India and many cricketers like Ajyinke Rahane, Virat Kohli , MS Dhoni are mentioned frequently
4. We could also say that links are shared very frequently on those tweets due to prominence of https in the word cloud.

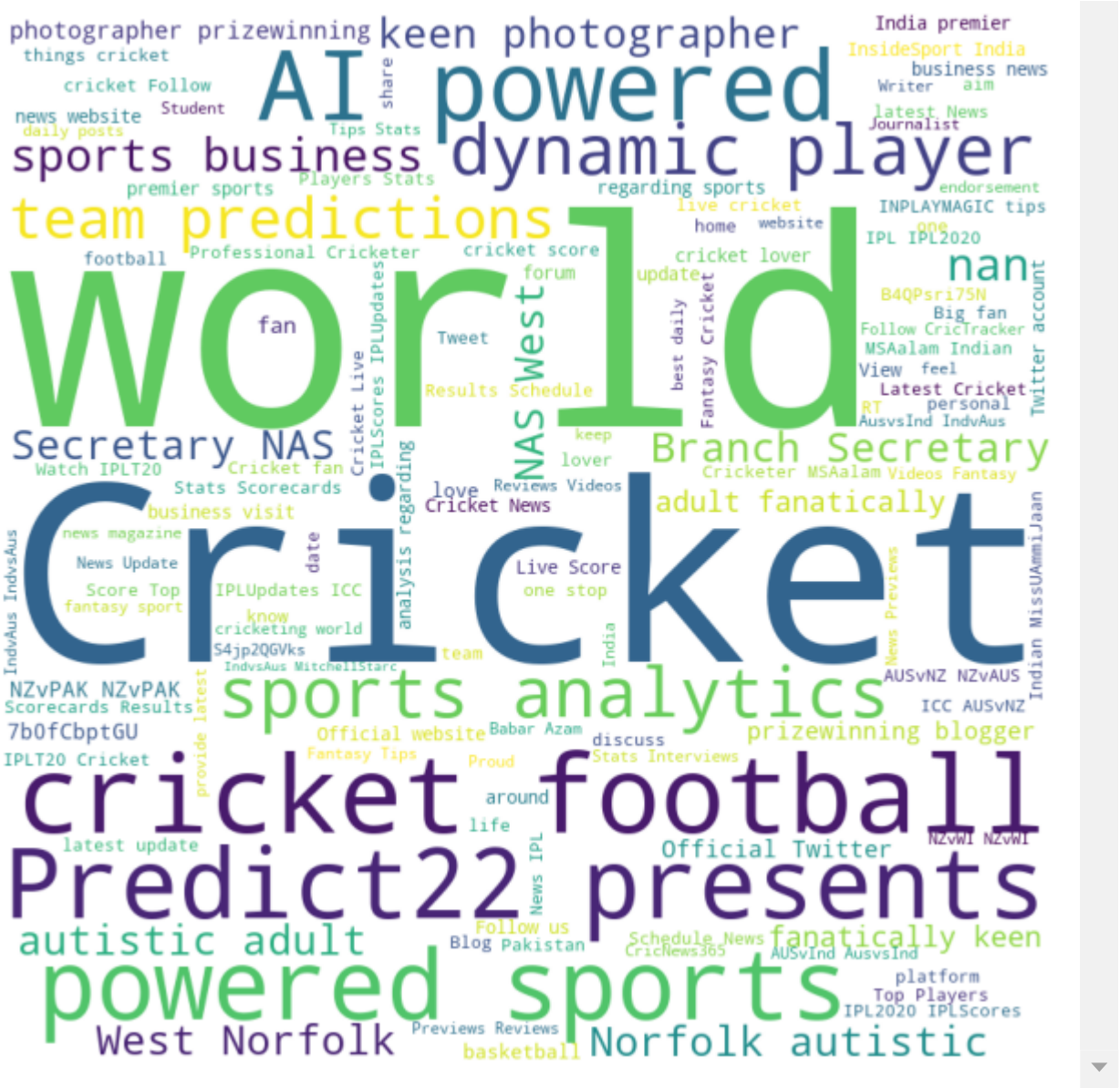
In [20]:
```
import os
import spacy
import io
from spacy.lang.en import English
# Use it on Jupyter Notebook or Google Colab
# DIR_PATH = os.getcwd()
# Use it on Python module
# DIR_PATH = os.path.dirname(__file__)

FILE_PATH = "tweets_info.csv"

import pandas as pd

# Read the file
df = pd.read_csv(FILE_PATH)
# Assign first_dialogue to the first row's "Dialogue" column
final_list = []
for i in range(10000):
    try:
        first_dialogue = df.loc[i, "user.description"]
        print(i)
        print(first_dialogue)
        # if first_dialogue == None:
        #   break;
        # use spacy with the dependency parse
        nlp = spacy.load("en_core_web_sm")
        #[str(sent) for sent in nlp(first_dialogue).sents]


        # use spacy with the sentencizer
        nlp = English()  # just the language with no model
        sentencizer = nlp.create_pipe("sentencizer")
        nlp.add_pipe(sentencizer)
        final_list.append(first_dialogue)
        # try:
        #   if (not(first_dialogue == nan)):
        #       k = [str(sent) for sent in nlp(first_dialogue).sents]
        #       print(k)
        #       for K in k:
        #           final_list.append(K)
        #       print(i)
        # except:
        #   print("Ok")
    except:
        continue;
with io.open("list_of_user_description.txt", "w", encoding="utf-8") as f:
    f.write((' '.join([str(elem) for elem in final_list])))
```

```
ian of The #Truth| #Film, #Cricket buff |#Champion #Winner T20FC | #BeHappy|
4989
#LiveTheSport
4990
🦁 Father, Husband, Engineer, Traveller, Investor and a Cricket fan!
4991
jazz musician ,trumpet  , biker ,barrister, ALP member . Australia 🔊 #invyh
```

```
Orhjaan band
4992
Your one-stop shop for fantasy sports previews, while also getting occasional
updates on Bollywood, politics, technology, health, lifestyle and more...
4993
Watch  #IPLT20 #Cricket🏏 Live Score, Top Players Stats, Scorecards, Result
s, Schedule & News. #IPL #IPL2020 #IPLScores #IPLUpdates  #ICC #AUSvNZ #NZvAU
S

4994
Sportslover 🏵
Independent political views 🙏
Always First Pakistan 🛡
```

Now we will make the word cloud.

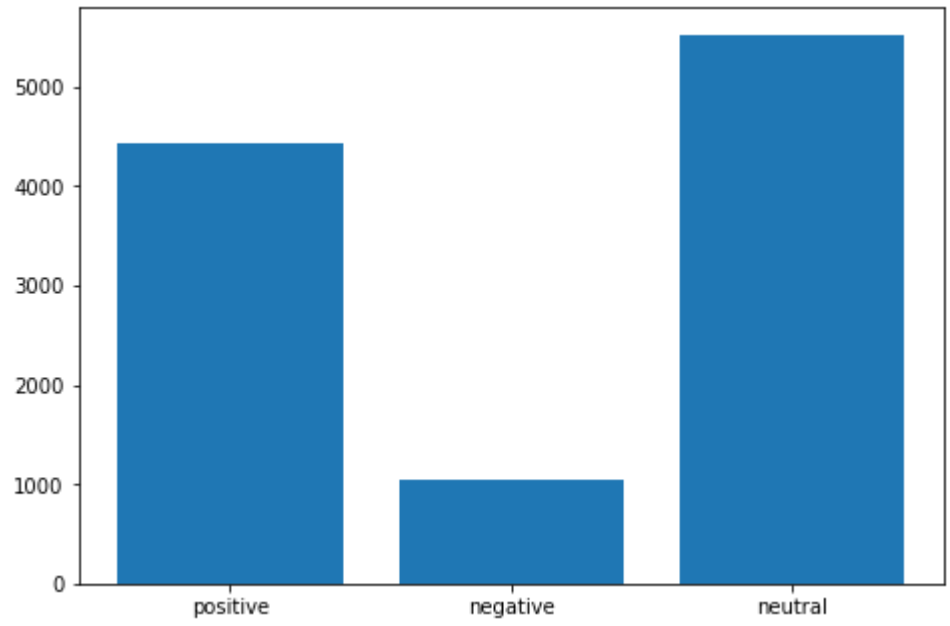
In [21]:
```
# Python program to generate WordCloud

# importing all necessery modules
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd
import io

# Reads 'Youtube04-Eminem.csv' file
with io.open("list_of_user_description.txt", "r", encoding="utf-8") as file:
    data = file.read().replace('\n', '')

comment_words = ''
STOPWORDS.add("https")
STOPWORDS.add("will")
STOPWORDS.add("t")
STOPWORDS.add("co")
STOPWORDS.add("Please")
stopwords = set(STOPWORDS)

# # iterate through the csv file
# for tokens in data:

#   tokens = tokens.lower()
#   print(tokens)

#   comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(data)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



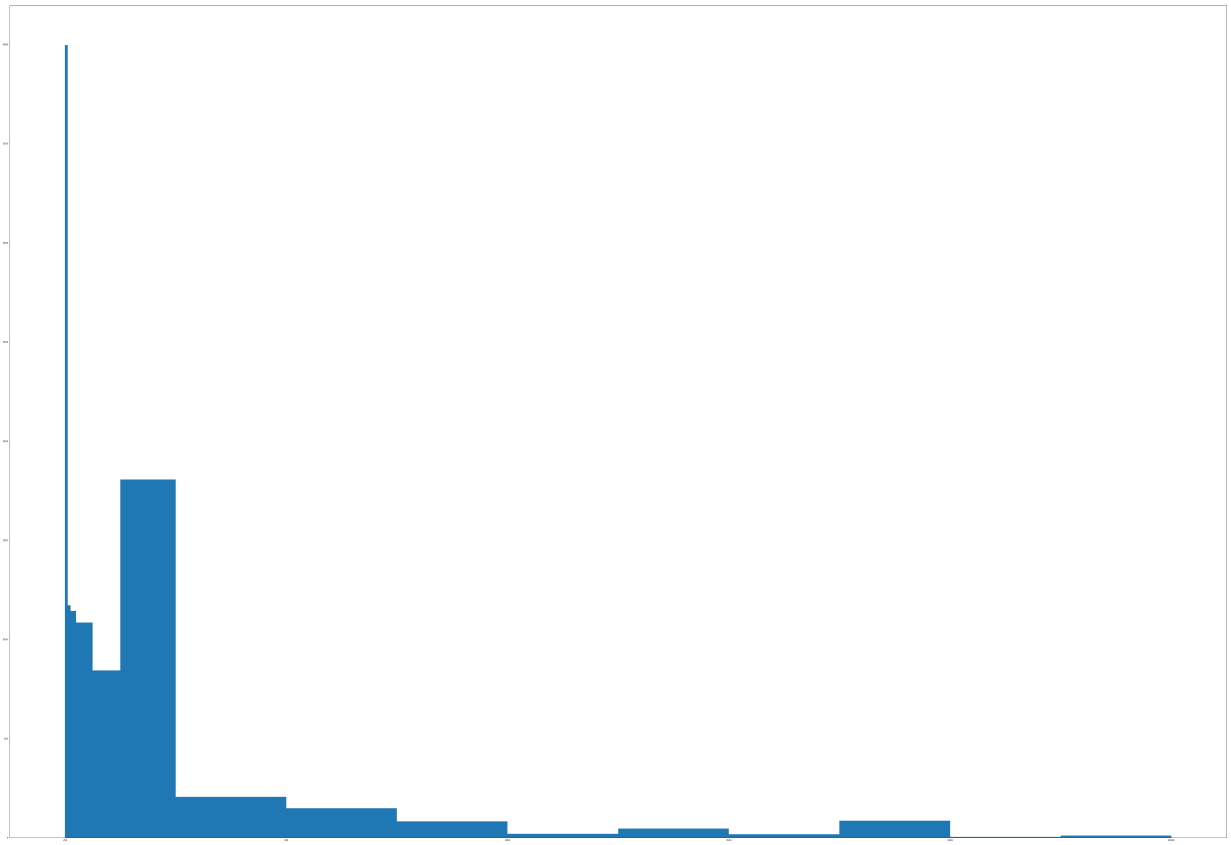
Now we will analyse the sentiment of the posts(positive, negative and neutral) and will present the findings with a bar chart. We have used a textblob module for this.

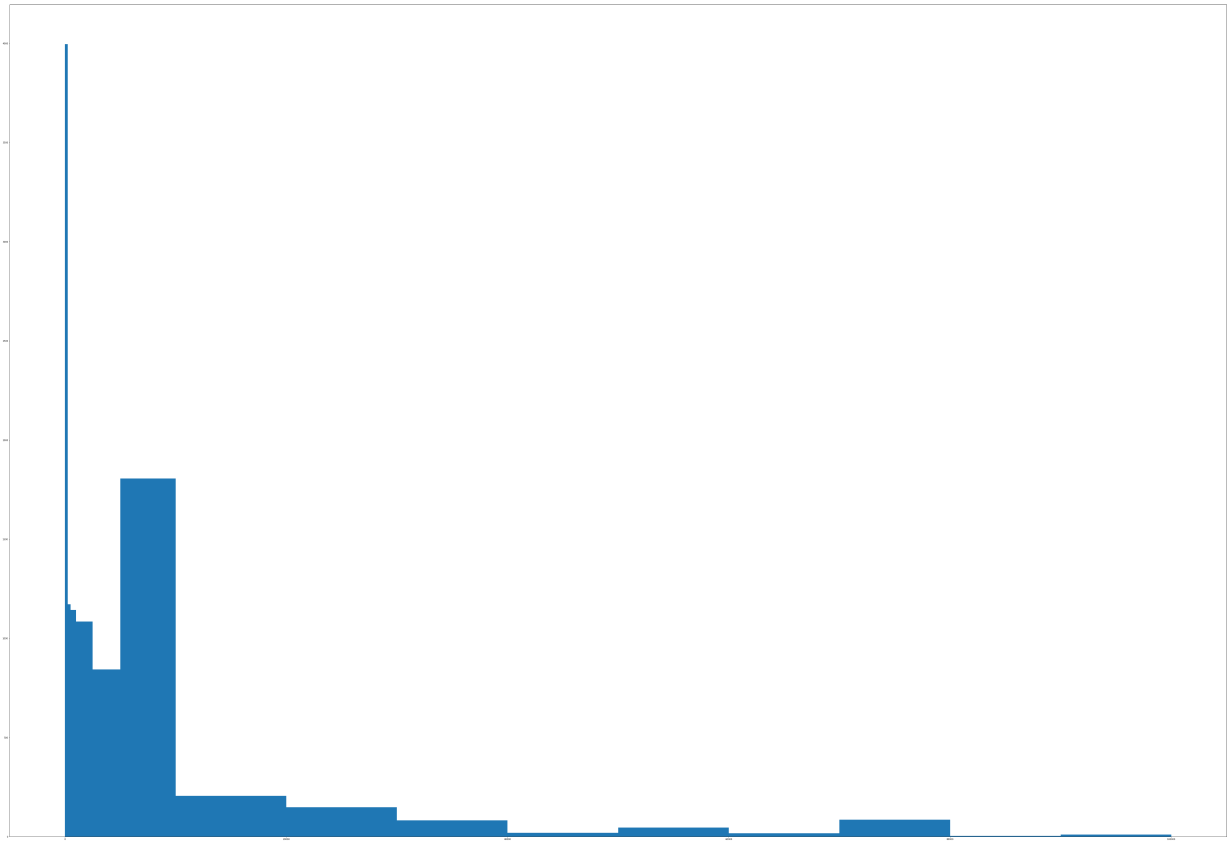
In [37]:
```
from textblob import TextBlob
import os
import spacy
import io
from spacy.lang.en import English
import matplotlib.pyplot as plt
# Use it on Jupyter Notebook or Google Colab
# DIR_PATH = os.getcwd()
# Use it on Python module


FILE_PATH =  "tweets_info.csv"

import pandas as pd

# Read the file
df = pd.read_csv(FILE_PATH)
# Assign first_dialogue to th
df["polarity"] =df['text'].apply(lambda tweet: TextBlob(tweet).polarity)

# print(df["polarity"])

positive  = 0
negative = 0
neutral = 0;

for dub in df["polarity"]:
#   print(dub)
    if dub > 0:
        positive += 1;
    if (dub< 0):
        negative += 1;
    if (dub == 0):
        neutral += 1;

label1 = ["positive", "negative", "neutral"]
label2 = [positive, negative, neutral]


fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
# ax.xlabel("Polarity")
ax.bar(label1, label2)
plt.show()

from matplotlib import pyplot as plt
import numpy as np


# Creating dataset
# a = np.array(df["user.followers_count"])

# # Creating histogram
# fig, ax = plt.subplots(figsize =(10, 7),
#                        tight_layout = True)
# ax.hist(a, bins = 1000 ,
```

```
#                     color='#607c8e')

# # Show plot
# plt.show()

a = np.array(df["user.followers_count"])

# print(df["user.followers_count"])
# Creating histogram
fig, ax = plt.subplots(figsize =(100, 70))
# cks = np.arange(0, 10000000, 100)x_ti
# plt.xticks(x_ticks)
ax.hist(a, bins = [0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000, 6
ax.set_xticklabels([0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000,
# ax.xlabel("Followers")

# ax.set_xlabel('marks')
# ax.set_ylabel('no. of students')


# Show plot
plt.show()

from matplotlib import pyplot as plt
import numpy as np


# Creating dataset
# a = np.array(df["user.followers_count"])

# # Creating histogram
# fig, ax = plt.subplots(figsize =(10, 7),
#                        tight_layout = True)
# ax.hist(a, bins = 1000 ,
#                     color='#607c8e')

# # Show plot
# plt.show()

a = np.array(df["user.followers_count"])

# print(df["user.friends_count"])
# Creating histogram
fig, ax = plt.subplots(figsize =(100, 70))
# cks = np.arange(0, 10000000, 100)x_ti
# plt.xticks(x_ticks)
ax.hist(a, bins = [0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000, 6
#ax.set_xticklabels([0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000,

# ax.xlabel("Friends")
# ax.set_xlabel('marks')
# ax.set_ylabel('no. of students')


# Show plot
plt.show()
```




```
<ipython-input-37-3aeb9c01ecb3>:70: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_xticklabels([0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 500
00, 60000, 70000, 80000, 90000, 100000], rotation=0, fontsize=10)
```





Now we will make a wordCloud of the tags associated with cricket that have been used.

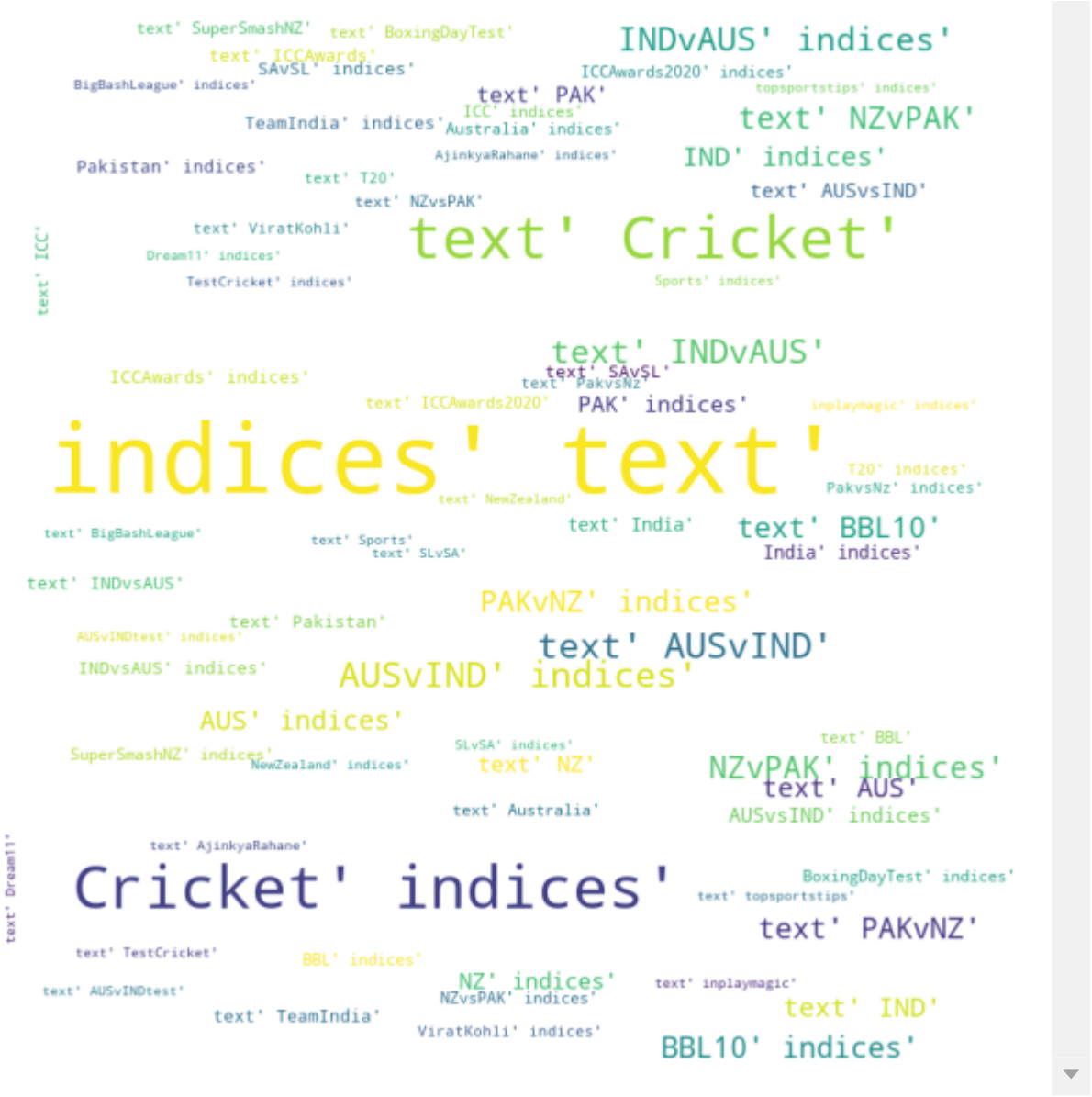
In [40]:
```
import os
import spacy
import io
from spacy.lang.en import English
# Use it on Jupyter Notebook or Google Colab
# DIR_PATH = os.getcwd()
# Use it on Python module

FILE_PATH =  "tweets_info.csv"

import pandas as pd

# Read the file
df = pd.read_csv(FILE_PATH)
# Assign first_dialogue to the first row's "Dialogue" column
final_list = []
for i in range(11000):
    print(i)
    try:
        first_dialogue = df.loc[i, "entities.hashtags"]
        print(i)
        print(first_dialogue)
        # if first_dialogue == None:
        #   break;
        # use spacy with the dependency parse
        nlp = spacy.load("en_core_web_sm")
        #[str(sent) for sent in nlp(first_dialogue).sents]


        # use spacy with the sentencizer
        nlp = English()  # just the language with no model
        sentencizer = nlp.create_pipe("sentencizer")
        nlp.add_pipe(sentencizer)
        final_list.append(first_dialogue)
    # try:
    #   if (not(first_dialogue == nan)):
    #       k = [str(sent) for sent in nlp(first_dialogue).sents]
    #       print(k)
    #       for K in k:
    #           final_list.append(K)
    #       print(i)
    except:
        print("Ok")
with io.open("list_of_hashtags_associated_with_tweets.txt", "w", encoding="utf-8"
    f.write((' '.join([str(elem) for elem in final_list])))
```

```
'AUSVIND', 'indices': [97, 105]}, {'text': 'cricket', 'indices': [106, 114]}]
10991
10991
[{'text': 'indiavsaustralia', 'indices': [32, 49]}, {'text': 'indvsaus', 'ind
ices': [50, 59]}, {'text': 'Test', 'indices': [60, 65]}, {'text': 'Match', 'i
ndices': [66, 72]}, {'text': 'jaspritbumrah', 'indices': [73, 87]}, {'text':
'Bowling', 'indices': [88, 96]}, {'text': 'Cricket', 'indices': [97, 105]}]
10992
10992
[{'text': 'bbl10', 'indices': [52, 58]}, {'text': 'bbl10', 'indices': [61, 6
7]}, {'text': 'fielding', 'indices': [68, 77]}, {'text': 'cricket', 'indice
```

```
s': [78, 86]}, {'text': 'T20', 'indices': [87, 91]}, {'text': 'BoxingDay', 'i
ndices': [92, 102]}]
10993
10993
[{'text': 'India', 'indices': [52, 58]}, {'text': 'Australia', 'indices': [5
9, 69]}, {'text': 'AUSvsIND', 'indices': [70, 79]}, {'text': 'Cricket', 'indi
ces': [80, 88]}, {'text': 'CricTracker', 'indices': [89, 101]}]
10994
10994
```

Now we need to use wordCloud.

In [41]:
```
# Python program to generate WordCloud

# importing all necessery modules
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd
import io

# Reads 'Youtube04-Eminem.csv' file
with io.open("list_of_hashtags_associated_with_tweets.txt", "r", encoding="utf-8"
    data = file.read().replace('\n', '')

comment_words = ''
STOPWORDS.add("https")
STOPWORDS.add("RahulGandhi")
STOPWORDS.add("Rahul Gandhi")
STOPWORDS.add("Rahul")
STOPWORDS.add("will")
STOPWORDS.add("t")
STOPWORDS.add("co")
STOPWORDS.add("Please")
STOPWORDS.add("text")
STOPWORDS.add("indices" '')
stopwords = set(STOPWORDS)

# # iterate through the csv file
# for tokens in data:

#   tokens = tokens.lower()
#   print(tokens)

#   comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(data)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Let's try to understand how often tweets related to cricket are retweeted, as while extracting the data, we were filtering out retweets to maintain uniqueness.

Now to analyse users, lets first understand what is the distrubution of number of followers each user has vs distribution of number of people being followed by the users.

In [42]:
```
from matplotlib import pyplot as plt
import numpy as np


# Creating dataset
# a = np.array(df["user.followers_count"])

# # Creating histogram
# fig, ax = plt.subplots(figsize =(10, 7),
#                        tight_layout = True)
# ax.hist(a, bins = 1000 ,
#                     color='#607c8e')

# # Show plot
# plt.show()

a = np.array(df["user.followers_count"])

print(df["user.followers_count"])
# Creating histogram
fig, ax = plt.subplots(figsize =(100, 70))
# cks = np.arange(0, 10000000, 100)x_ti
# plt.xticks(x_ticks)
ax.hist(a, bins = [0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000, 6
ax.set_xticklabels([0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000,


# ax.set_xlabel('marks')
# ax.set_ylabel('no. of students')


# Show plot
plt.show()

from matplotlib import pyplot as plt
import numpy as np


# Creating dataset
# a = np.array(df["user.followers_count"])

# # Creating histogram
# fig, ax = plt.subplots(figsize =(10, 7),
#                        tight_layout = True)
# ax.hist(a, bins = 1000 ,
#                     color='#607c8e')

# # Show plot
# plt.show()

a = np.array(df["user.followers_count"])

print(df["user.friends_count"])
# Creating histogram
fig, ax = plt.subplots(figsize =(100, 70))
# cks = np.arange(0, 10000000, 100)x_ti
```

```
# plt.xticks(x_ticks)
ax.hist(a, bins = [0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000, 6
ax.set_xticklabels([0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 50000,


# ax.set_xlabel('marks')
# ax.set_ylabel('no. of students')


# Show plot
plt.show()
```

```
0         809
1        4996
2         694
3        1237
4         809
        ...
10994     549
10995      83
10996    1553
10997     528
10998     156
Name: user.friends_count, Length: 10999, dtype: int64

<ipython-input-42-a616788fb2cd>:59: UserWarning: FixedFormatter should only b
e used together with FixedLocator
  ax.set_xticklabels([0, 250, 500, 1000,2500, 5000,10000,20000,30000,40000, 5
0000, 60000, 70000, 80000, 90000, 100000], rotation=0, fontsize=10)
```

Make a word cloud of description of users to understand what walks of life cricket tweeters come from :D.

In [45]:
```
import os
import spacy
import io
from spacy.lang.en import English
# Use it on Jupyter Notebook or Google Colab
# DIR_PATH = os.getcwd()
# Use it on Python module


FILE_PATH =  "tweets_info.csv"

import pandas as pd

# Read the file
df = pd.read_csv(FILE_PATH)
# Assign first_dialogue to the first row's "Dialogue" column
final_list = []
for i in range(1000):
    first_dialogue = df.loc[i, "user.description"]
    print(i)
    print(first_dialogue)
    # if first_dialogue == None:
    #   break;
    # use spacy with the dependency parse
    nlp = spacy.load("en_core_web_sm")
    #[str(sent) for sent in nlp(first_dialogue).sents]


    # use spacy with the sentencizer
    nlp = English()  # just the language with no model
    sentencizer = nlp.create_pipe("sentencizer")
    nlp.add_pipe(sentencizer)
    final_list.append(first_dialogue)
    # try:
    #   if (not(first_dialogue == nan)):
    #       k = [str(sent) for sent in nlp(first_dialogue).sents]
    #       print(k)
    #       for K in k:
    #           final_list.append(K)
    #       print(i)
    # except:
    #   print("Ok")
with io.open("list_of_user_description.txt", "w", encoding="utf-8") as f:
    f.write((' '.join([str(elem) for elem in final_list])))
```

190+ Subscribers

```
1961 Subscribers
32
#TeamIndia Fan IN; MSDian 😍; Hungry for Cricket 🏏
33
Official website of InsideSport - India's premier sports business news websit
e. For news and analysis regarding sports business,visit https://t.co/B4QPsri
75N (https://t.co/B4QPsri75N)
34
Consultant (@HumphreysAdvice / @DebtWolf) | Strategist | Litigation Funder |
```

In [46]:
```
# Python program to generate WordCloud

# importing all necessery modules
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd
import io

# Reads 'Youtube04-Eminem.csv' file
with io.open("list_of_user_description.txt", "r", encoding="utf-8") as file:
    data = file.read().replace('\n', '')

comment_words = ''
STOPWORDS.add("https")
STOPWORDS.add("RahulGandhi")
STOPWORDS.add("Rahul Gandhi")
STOPWORDS.add("Rahul")
STOPWORDS.add("will")
STOPWORDS.add("t")
STOPWORDS.add("co")
STOPWORDS.add("Please")
stopwords = set(STOPWORDS)

# # iterate through the csv file
# for tokens in data:

#   tokens = tokens.lower()
#   print(tokens)

#   comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(data)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Lers plot the retweeting frequency.

In [ ]: