



WhatsApp is a great source of data to analyze many patterns and relationships between two or more people chatting personally or even in groups. If you want to know how we can analyze the sentiments of a WhatsApp chat, this Code is for you

WhatsApp Chat Sentiment Analysis To analyze the sentiments of a WhatsApp chat, we need to collect data from WhatsApp. Most of you must be using this messaging app, so to collect data about your chat, simply follow the steps mentioned below: For iPhone: Open your chat with a person or a group Just tap on the profile of the person or the group You will see an option to export chat down below For Android: Open your chat with a person or a group Click on the three dots above Click on more Click on the export chat You will see an option to attach media while exporting your chat. For simplicity, it is best not to attach media. Finally, enter your email and you will find your WhatsApp chat in your inbox. Now let's start with the task of WhatsApp chat sentiment analysis with Python. I'll start this task by defining some helper functions because the data we get from WhatsApp is not a dataset that is ready to be used for any kind of data science task. So, to prepare your data for the sentiment analysis task, just define all the functions as defined below:

```
In [ ]: import re
import pandas as pd
import numpy as np
import emoji
from collections import Counter
import matplotlib.pyplot as plt
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

# Extract Time
def date_time(s):
    pattern = '^([0-9]+)(\\/)([0-9]+)(\\/)([0-9]+), ([0-9]+):([0-9]+)[ ]?(AM|PM|am|pm)? -'
    result = re.match(pattern, s)
    if result:
```

```

        return True
    return False

# Find Authors or Contacts
def find_author(s):
    s = s.split(":")
    if len(s)==2:
        return True
    else:
        return False

# Finding Messages
def getDatapoint(line):
    splitline = line.split(' - ')
    dateTime = splitline[0]
    date, time = dateTime.split(", ")
    message = " ".join(splitline[1:])
    if find_author(message):
        splitmessage = message.split(": ")
        author = splitmessage[0]
        message = " ".join(splitmessage[1:])
    else:
        author= None
    return date, time, author, message

```

It doesn't matter if you are using a group chat dataset or your conversation with one person. All the functions defined above will prepare your data for the task of sentiment analysis as well as for any data science task. Now here is how we can prepare the data we collected from WhatsApp by using the above functions:

```

In [ ]:
data = []
conversation = 'WhatsApp Chat with Sapna.txt'
with open(conversation, encoding="utf-8") as fp:
    fp.readline()
    messageBuffer = []
    date, time, author = None, None, None
    while True:
        line = fp.readline()
        if not line:
            break
        line = line.strip()
        if date_time(line):
            if len(messageBuffer) > 0:
                data.append([date, time, author, ' '.join(messageBuffer)])
                messageBuffer.clear()
            date, time, author, message = getDatapoint(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)

```

In []: Now here **is** how we can analyze the sentiments of WhatsApp chat using Python:

```

In [ ]:
df = pd.DataFrame(data, columns=["Date", 'Time', 'Author', 'Message'])
df['Date'] = pd.to_datetime(df['Date'])

data = df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in data["Message"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in data["Message"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in data["Message"]]
print(data.head())

```

Date Time Author ... Positive Negative Neutral 0 2020-04-06 12:30 pm Sapna ... 0.0 0.000 1.000 1 2020-04-06 12:30 pm Sapna ... 0.0 0.000 1.000 2 2020-04-06 12:54 pm Aman Kharwal ... 0.0 0.000 1.000 3 2020-04-06 12:55 pm Sapna ... 0.0 0.383 0.617 4 2020-04-06 1:00 pm Aman Kharwal ... 0.0 0.000 1.000

```
In [ ]: x = sum(data["Positive"])
        y = sum(data["Negative"])
        z = sum(data["Neutral"])

        def sentiment_score(a, b, c):
            if (a>b) and (a>c):
                print("Positive 😊 ")
            elif (b>a) and (b>c):
                print("Negative 😞 ")
            else:
                print("Neutral 😐 ")
        sentiment_score(x, y, z)
```

Output: Neutral 😐

So, the data I used indicates that most of the messages between me and the other person are neutral. Which means it's neither positive nor negative.

So this is how we can perform the task of sentiment analysis of WhatsApp chat. WhatsApp is a great source of data for the task of sentiment analysis and every data science task based on natural language processing.

In []: