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
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
    else:
        return False
def find_contact(s):
    s=s.split(":")
    if len(s)==2:
        return True
    else:
        return False
def getMassage(line):
    splitline=line.split(' - ')
    datetime=splitline[0];
    date,time=datetime.split(', ')
    message=" ".join(splitline[1:])
    if find_contact(message):
        splitmessage=message.split(": ")
        author=splitmessage[0]
        message=" ".join(splitline[1:])
    else:
        author=None
    return date,time,author,message
data=[]
conversation="chat.txt.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=getMassage(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)
df=pd.DataFrame(date,columns=["Date", "Time","Contact","Message"])
df['Date']=pd.to_datetime(df["Date"])
date=df.dropna()
from nltk.sentiment.vader import sentimentIntensityAnalyzer
sentiments=SentimentIntensityAnalyzer()
date["positive"]=[sentiments.polarity_scores(i)["pos"] for i in data["Message"]]
date["negative"]=[sentiments.polarity_scores(i)["neg"] for i in data["Message"]]
date["neutral"]=[sentiments.polarity_scores(i)["neu"] for i in data["Message"]]
date.head(20)
```