In [16]:
```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
#from sklearn.model_selection import train_test_split
#from sklearn.metrics import classification_report
#from sklearn.naive_bayes import GaussianNB
#from sklearn.naive_bayes import CategoricalNB
#from sklearn.naive_bayes import BernoulliNB
warnings.filterwarnings('ignore')
```

In [6]:
```
# load data
df = pd.read_csv('MERGETWEETS.CSV', error_bad_lines=False, delimiter=";")
```

In [7]:
```
# read data
df.head()
```

Out[7]:

	Date	User	Tweet	Sentimen
0	2021-05-24 14:48:08	samaninatasha	RT @omar_quraishi: 2 million doses of the Covi...	netral
1	2021-05-24 14:47:32	AeruSenpai	RT @azrulmohdkhalib: Ppl who know our work kno...	netral
2	2021-05-24 14:47:28	bawany777	RT @OfficialNcoc: PIA plane carrying 2 Million...	netral
3	2021-05-24 14:46:49	omar_quraishi	RT @Urwah43385573: I again went today and ther...	netral
4	2021-05-24 14:46:15	meshacassie	RT @boosulyn: Pharmaniaga does not have a mono...	netral

In [8]:
```
# Melihat isi kolom 'Tweet'
df['Tweet']
```

Out[8]:
```
0        RT @omar_quraishi: 2 million doses of the Covi...
1        RT @azrulmohdkhalib: Ppl who know our work kno...
2        RT @OfficialNcoc: PIA plane carrying 2 Million...
3        RT @Urwah43385573: I again went today and ther...
4        RT @boosulyn: Pharmaniaga does not have a mono...
                               ...
77558    RT @business: An investor in Sinovac reports a...
77559    An investor in Sinovac reports a nearly six-fo...
77560    An investor in Sinovac reports a nearly six-fo...
77561    An investor in Sinovac reports a nearly six-fo...
77562                                                  NaN
Name: Tweet, Length: 77563, dtype: object
```

In [13]:
```
# Melihat isi tweet baris 1
df.iloc[:1, 2:3].values
```

Out[13]:
```
array([['RT @omar_quraishi: 2 million doses of the Covid-19 vaccine Sinovac r
each Pakistan https://t.co/m3LAYd9pDO']],
      dtype=object)
```

In [17]:
```
import neattext.functions as nfx
```

In [18]:
```
#methods
#dir(nfx)
```

In [22]:
```
#print(df['Tweet'])
from neattext import TextExtractor
def removeHashtag(s):
    docx = TextExtractor()
    docx.text = s
    docx.extract_hashtags()
    return docx
```

In [23]:
```
import neattext.functions as nfx
df["Tweet"].apply(lambda x : nfx.remove_hashtags(str(x)))
```

Out[23]:
```
0        RT @omar_quraishi: 2 million doses of the Covi...
1        RT @azrulmohdkhalib: Ppl who know our work kno...
2        RT @OfficialNcoc: PIA plane carrying 2 Million...
3        RT @Urwah43385573: I again went today and ther...
4        RT @boosulyn: Pharmaniaga does not have a mono...
                               ...
77558    RT @business: An investor in Sinovac reports a...
77559    An investor in Sinovac reports a nearly six-fo...
77560    An investor in Sinovac reports a nearly six-fo...
77561    An investor in Sinovac reports a nearly six-fo...
77562                                                  nan
Name: Tweet, Length: 77563, dtype: object
```

In [24]:
```
df['extracted_hastags'] = df['Tweet'].apply(lambda x: nfx.extract_hashtags(st
print (df['extracted_hastags'])
```

```
0                              []
1        [#Pharmaniaga, #COVID19]
2                              []
3                              []
4                              []
                  ...
77558                          []
77559                          []
77560                          []
77561                          []
77562                          []
Name: extracted_hastags, Length: 77563, dtype: object
```

In [25]:
```
#clean tweet
df['clean_tweet'] = df['Tweet'].apply(lambda x: nfx.remove_hashtags(str(x)))
df[['Tweet', 'clean_tweet']]
```

Out[25]:

	Tweet	clean_tweet
0	RT @omar_quraishi: 2 million doses of the Covi...	RT @omar_quraishi: 2 million doses of the Covi...
1	RT @azrulmohdkhalib: Ppl who know our work kno...	RT @azrulmohdkhalib: Ppl who know our work kno...
2	RT @OfficialNcoc: PIA plane carrying 2 Million...	RT @OfficialNcoc: PIA plane carrying 2 Million...
3	RT @Urwah43385573: I again went today and ther...	RT @Urwah43385573: I again went today and ther...

	Tweet	clean_tweet
4	RT @boosulyn: Pharmaniaga does not have a mono...	RT @boosulyn: Pharmaniaga does not have a mono...
...	...	...
77558	RT @business: An investor in Sinovac reports a...	RT @business: An investor in Sinovac reports a...
77559	An investor in Sinovac reports a nearly six-fo...	An investor in Sinovac reports a nearly six-fo...
77560	An investor in Sinovac reports a nearly six-fo...	An investor in Sinovac reports a nearly six-fo...
77561	An investor in Sinovac reports a nearly six-fo...	An investor in Sinovac reports a nearly six-fo...
77562	NaN	nan

77563 rows × 2 columns

In [26]:
```
#remove user handle
df['clean_tweet'] = df['clean_tweet'].apply(lambda x: nfx.remove_userhandles(
df[['Tweet', 'clean_tweet']]
```

Out[26]:

	Tweet	clean_tweet
0	RT @omar_quraishi: 2 million doses of the Covi...	RT 2 million doses of the Covid-19 vaccine S...
1	RT @azrulmohdkhalib: Ppl who know our work kno...	RT Ppl who know our work know tht I am very ...
2	RT @OfficialNcoc: PIA plane carrying 2 Million...	RT PIA plane carrying 2 Million doses of Sin...
3	RT @Urwah43385573: I again went today and ther...	RT I again went today and there was a doctor...
4	RT @boosulyn: Pharmaniaga does not have a mono...	RT Pharmaniaga does not have a monopoly over...
...	...	...
77558	RT @business: An investor in Sinovac reports a...	RT An investor in Sinovac reports a nearly s...
77559	An investor in Sinovac reports a nearly six-fo...	An investor in Sinovac reports a nearly six-fo...
77560	An investor in Sinovac reports a nearly six-fo...	An investor in Sinovac reports a nearly six-fo...
77561	An investor in Sinovac reports a nearly six-fo...	An investor in Sinovac reports a nearly six-fo...
77562	NaN	nan

77563 rows × 2 columns

In [27]:
```
df['clean_tweet'].iloc[10]
```

Out[27]:
```
'RT   With 44% of its population fully vaccinated, Bahrain must be rather fru
strated by the performance of the Sinovac vaccin…'
```

In [28]:
```
#cleaning text remove multiple white spaces
df['clean_tweet'] = df['clean_tweet'].apply(nfx.remove_multiple_spaces)
df['clean_tweet'].iloc[10]
```

Out[28]:
```
'RT With 44% of its population fully vaccinated, Bahrain must be rather frust
rated by the performance of the Sinovac vaccin…'
```

In [29]:
```
#remove urls
```

```
df['clean_tweet'] = df['clean_tweet'].apply(nfx.remove_urls)
df['clean_tweet']
```

Out[29]:
```
0        RT 2 million doses of the Covid-19 vaccine Sin...
1        RT Ppl who know our work know tht I am very cr...
2        RT PIA plane carrying 2 Million doses of Sinov...
3        RT I again went today and there was a doctor i...
4        RT Pharmaniaga does not have a monopoly over C...
                              ...
77558    RT An investor in Sinovac reports a nearly six...
77559    An investor in Sinovac reports a nearly six-fo...
77560    An investor in Sinovac reports a nearly six-fo...
77561    An investor in Sinovac reports a nearly six-fo...
77562                                                  nan
Name: clean_tweet, Length: 77563, dtype: object
```

In [31]:
```
#remove punctuations-tanda baca
df['clean_tweet'] = df['clean_tweet'].apply(nfx.remove_puncts)
df['clean_tweet']
```

Out[31]:
```
0        RT 2 million doses of the Covid19 vaccine Sino...
1        RT Ppl who know our work know tht I am very cr...
2        RT PIA plane carrying 2 Million doses of Sinov...
3        RT I again went today and there was a doctor i...
4        RT Pharmaniaga does not have a monopoly over C...
                              ...
77558    RT An investor in Sinovac reports a nearly six...
77559    An investor in Sinovac reports a nearly sixfol...
77560    An investor in Sinovac reports a nearly sixfol...
77561    An investor in Sinovac reports a nearly sixfol...
77562                                                  nan
Name: clean_tweet, Length: 77563, dtype: object
```

In [32]:
```
#remove emojis
df['clean_tweet'] = df['clean_tweet'].apply(nfx.replace_emojis)
```

In [33]:
```
#remove special character
df['clean_tweet'] = df['clean_tweet'].apply(nfx.remove_special_characters)
df['clean_tweet']
```

Out[33]:
```
0        RT 2 million doses of the Covid19 vaccine Sino...
1        RT Ppl who know our work know tht I am very cr...
2        RT PIA plane carrying 2 Million doses of Sinov...
3        RT I again went today and there was a doctor i...
4        RT Pharmaniaga does not have a monopoly over C...
                              ...
77558    RT An investor in Sinovac reports a nearly six...
77559    An investor in Sinovac reports a nearly sixfol...
77560    An investor in Sinovac reports a nearly sixfol...
77561    An investor in Sinovac reports a nearly sixfol...
77562                                                  nan
Name: clean_tweet, Length: 77563, dtype: object
```

In [44]:
```
#sentiment analysis
from textblob import TextBlob
def get_sentiment(Tweet):
  blob = TextBlob(Tweet)
  sentiment_polarity = blob.sentiment.polarity
  sentiment_subjectivity = blob.sentiment.subjectivity
  if sentiment_polarity > 0:
    sentiment_label = 'positive'
  elif sentiment_polarity < 0:
    sentiment_label = 'negative'
```

```
  else:
    sentiment_label = 'neutral'
  result = {'data_polarity': sentiment_polarity,
            'data_subjectivity': sentiment_subjectivity,
            'data_sentiment': sentiment_label}
  return result

exl = df['clean_tweet'].iloc[123]

get_sentiment(exl)
```

Out[44]:
```
{'data_polarity': 0.0, 'data_subjectivity': 0.0, 'data_sentiment': 'neutral'}
```

In [45]:
```
df['sentiment_results'] = df['clean_tweet'].apply(get_sentiment)
df['sentiment_results']
```

Out[45]:
```
0        {'data_polarity': 0.0, 'data_subjectivity': 0....
1        {'data_polarity': 0.0, 'data_subjectivity': 0....
2        {'data_polarity': 0.0, 'data_subjectivity': 0....
3        {'data_polarity': 0.0, 'data_subjectivity': 0....
4        {'data_polarity': 0.0, 'data_subjectivity': 0....
                              ...
77558    {'data_polarity': 0.175, 'data_subjectivity': ...
77559    {'data_polarity': 0.175, 'data_subjectivity': ...
77560    {'data_polarity': 0.175, 'data_subjectivity': ...
77561    {'data_polarity': 0.175, 'data_subjectivity': ...
77562    {'data_polarity': 0.0, 'data_subjectivity': 0....
Name: sentiment_results, Length: 77563, dtype: object
```

In [46]:
```
#split
df['sentiment_results'].iloc[77558]
```

Out[46]:
```
{'data_polarity': 0.175,
 'data_subjectivity': 0.3666666666666667,
 'data_sentiment': 'positive'}
```

In [47]:
```
#normalize
pd.json_normalize(df['sentiment_results'].iloc[77558])
```

Out[47]:

	data_polarity	data_subjectivity	data_sentiment
0	0.175	0.366667	positive

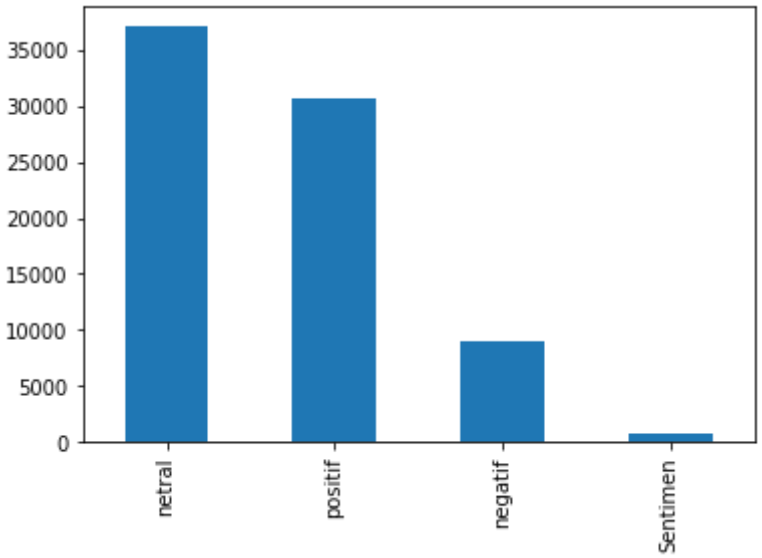
In [48]:
```
df['sentiment_results']
```

Out[48]:
```
0        {'data_polarity': 0.0, 'data_subjectivity': 0....
1        {'data_polarity': 0.0, 'data_subjectivity': 0....
2        {'data_polarity': 0.0, 'data_subjectivity': 0....
3        {'data_polarity': 0.0, 'data_subjectivity': 0....
4        {'data_polarity': 0.0, 'data_subjectivity': 0....
                              ...
77558    {'data_polarity': 0.175, 'data_subjectivity': ...
77559    {'data_polarity': 0.175, 'data_subjectivity': ...
77560    {'data_polarity': 0.175, 'data_subjectivity': ...
77561    {'data_polarity': 0.175, 'data_subjectivity': ...
77562    {'data_polarity': 0.0, 'data_subjectivity': 0....
Name: sentiment_results, Length: 77563, dtype: object
```

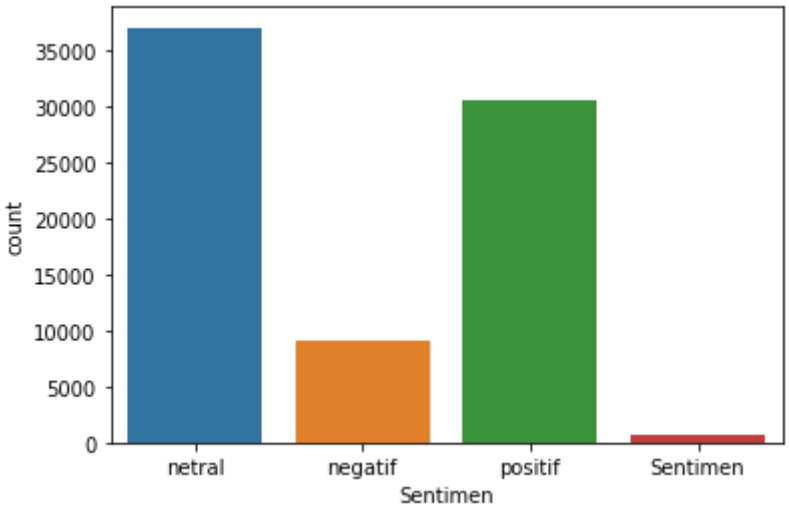
In [49]:
```
import pandas as pd
df = df.join(pd.json_normalize(df['sentiment_results']),how='left')
```

```
#df.head()
```

In [51]:
```
df['Sentimen'].value_counts()
import matplotlib.pyplot
df['Sentimen'].value_counts().plot(kind='bar')
plt.show()
```



In [53]:
```
#plot with seaborn
sns.countplot(df['Sentimen'])
plt.show()
```



In [54]:
```
#keyword extraction for positive , negative, and neutral
positive_tweet = df[df['Sentimen'] == 'positif']['clean_tweet']
```

In [55]:
```
negative_tweet = df[df['Sentimen'] == 'negatif']['clean_tweet']
```

In [56]:
```
neutral_tweet = df[df['Sentimen'] == 'netral']['clean_tweet']
```

In [57]:
```
#neutral_tweet
```

In [58]:
```
#negative_tweet
```

In [59]:
```
#positive_tweet
```

In [60]:
```
#remove stopwords and convert to tekns
positive_tweet_list=positive_tweet.apply(nfx.remove_stopwords).tolist()
```

In [61]:
```
#positive_tweet_list
```

In [62]:
```
negative_tweet_list=negative_tweet.apply(nfx.remove_stopwords).tolist()
```

In [63]:
```
neutral_tweet_list=neutral_tweet.apply(nfx.remove_stopwords).tolist()
```

In [64]:
```
#neutral_tweet_list
```

In [65]:
```
#negative_tweet_list
```

In [66]:
```
#tokenize
#for line in positive_tweet_list:
  #print(line)
  #for token in line.split():
      #print (token)
```

In [67]:
```
pos_tokens = [token for line in positive_tweet_list for token in line.split()
```

In [68]:
```
neg_tokens = [token for line in negative_tweet_list for token in line.split()
```

In [69]:
```
neut_tokens = [token for line in neutral_tweet_list for token in line.split()
```

In [70]:
```
#get most communest keyword
from collections import Counter
```

In [71]:
```
def get_tokens(docx,num=30):
    word_tokens = Counter(docx)
    most_common = word_tokens.most_common(num)
    result = dict(most_common)
    return result
```

In [72]:
```
#get_tokens(pos_tokens)
```

In [73]:
```
most_common_pos_words = get_tokens(pos_tokens)
```

In [74]:
```
most_common_neg_words = get_tokens(neg_tokens)
```

In [75]:
```
most_common_neut_words = get_tokens(neut_tokens)
```

In [76]:
```
#plot with seaborn
neg_df = pd.DataFrame(most_common_neg_words.items(),columns=['words','scores'
```

In [77]:
```
#plot positive
pos_df = pd.DataFrame(most_common_pos_words.items(),columns=['words', 'scores
```

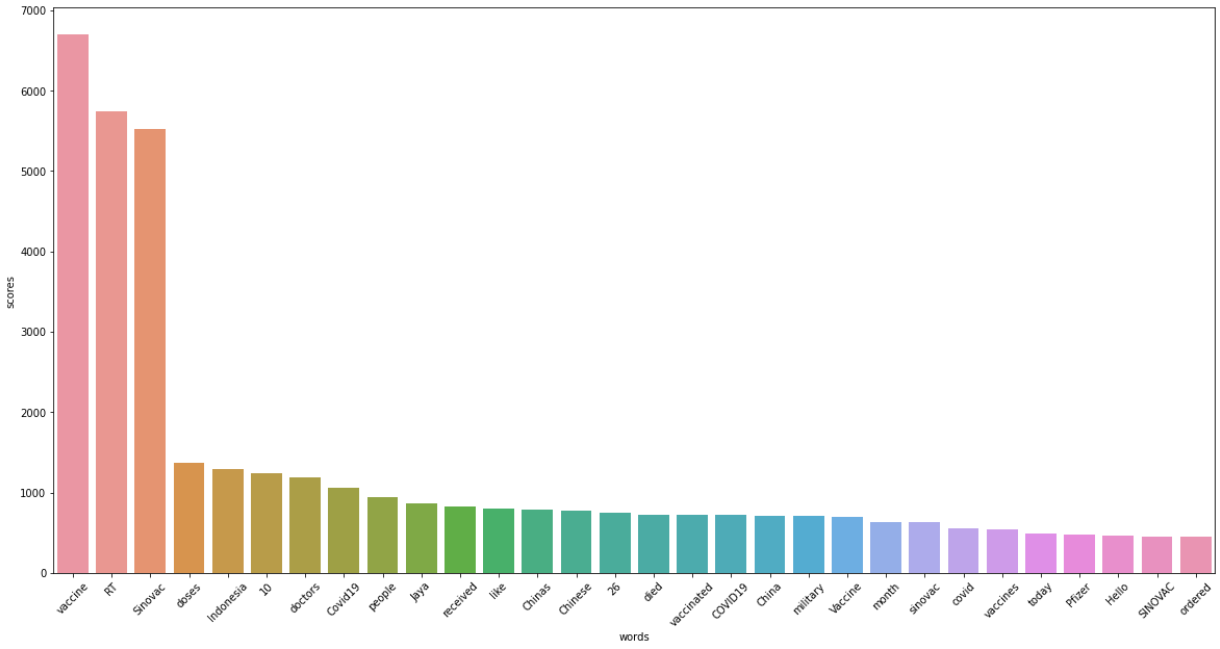
In [78]:
```
#plot neutral
neut_df = pd.DataFrame(most_common_neut_words.items(),columns=['words', 'scor
```

In [80]:
```
#pos_df
```

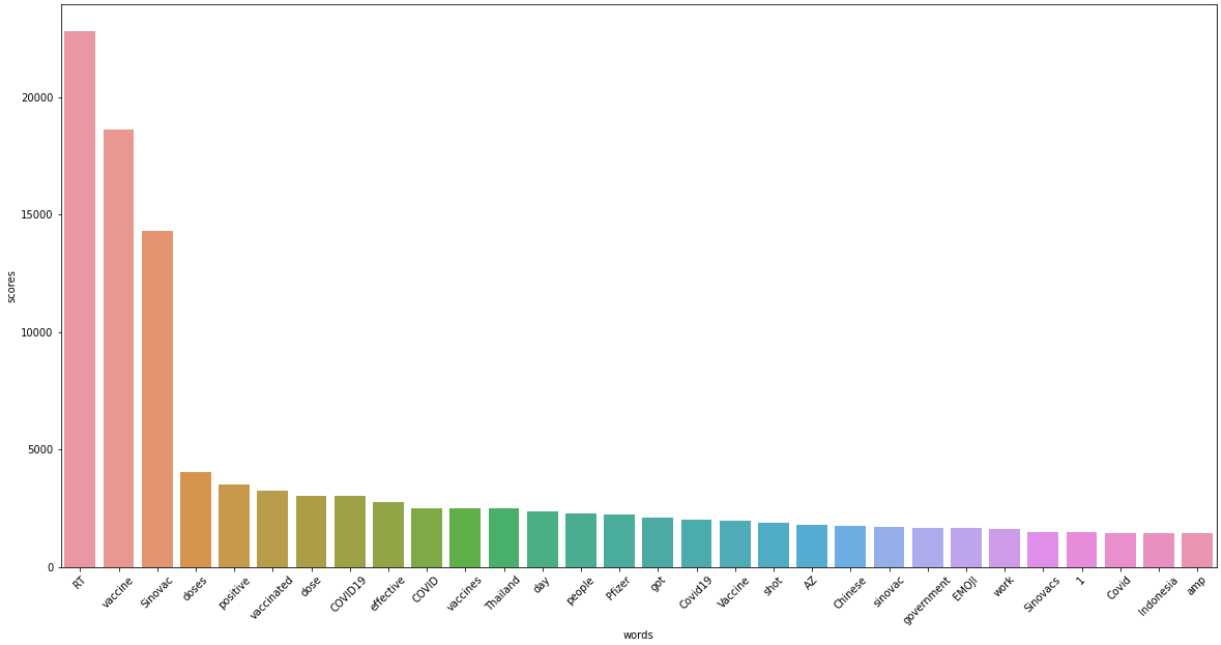
In [81]:
```
#neut_df
```

In [82]:
```
#neg_df
```

In [83]:
```
plt.figure(figsize=(20,10))
sns.barplot(x='words', y='scores', data =neg_df)
plt.xticks(rotation=45)
plt.show()
```

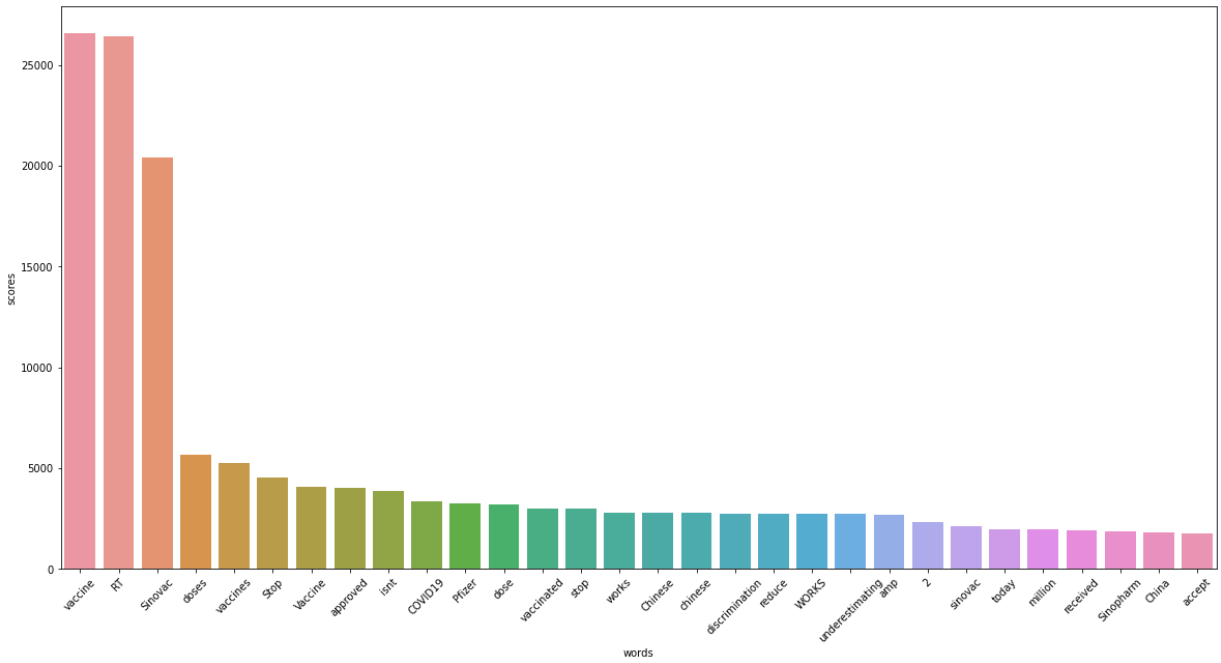


In [84]:
```
pos_df = pd.DataFrame(most_common_pos_words.items(),columns=['words', 'scores
plt.figure(figsize=(20,10))
sns.barplot(x='words', y='scores', data=pos_df)
plt.xticks(rotation=45)
plt.show()
```



In [85]:

```
neut_df = pd.DataFrame(most_common_neut_words.items(),columns=['words', 'scor
plt.figure(figsize=(20,10))
sns.barplot(x='words', y='scores', data=neut_df)
plt.xticks(rotation=45)
plt.show()
```



In [86]:

```
#world cloud
!pip install WordCloud
from wordcloud import WordCloud
```

```
Requirement already satisfied: WordCloud in /home/harysabita/anaconda3/lib/py
thon3.8/site-packages (1.8.1)
Requirement already satisfied: numpy>=1.6.1 in /home/harysabita/anaconda3/li
b/python3.8/site-packages (from WordCloud) (1.19.5)
Requirement already satisfied: pillow in /home/harysabita/anaconda3/lib/pytho
n3.8/site-packages (from WordCloud) (8.2.0)
Requirement already satisfied: matplotlib in /home/harysabita/anaconda3/lib/p
ython3.8/site-packages (from WordCloud) (3.4.3)
Requirement already satisfied: cycler>=0.10 in /home/harysabita/anaconda3/li
b/python3.8/site-packages (from matplotlib->WordCloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/harysabita/anaconda
```

```
3/lib/python3.8/site-packages (from matplotlib->WordCloud) (1.3.1)
Requirement already satisfied: python-dateutil>=2.7 in /home/harysabita/anaco
nda3/lib/python3.8/site-packages (from matplotlib->WordCloud) (2.8.1)
Requirement already satisfied: pyparsing>=2.2.1 in /home/harysabita/anaconda
3/lib/python3.8/site-packages (from matplotlib->WordCloud) (2.4.7)
Requirement already satisfied: six in /home/harysabita/anaconda3/lib/python3.
8/site-packages (from cycler>=0.10->matplotlib->WordCloud) (1.15.0)
```

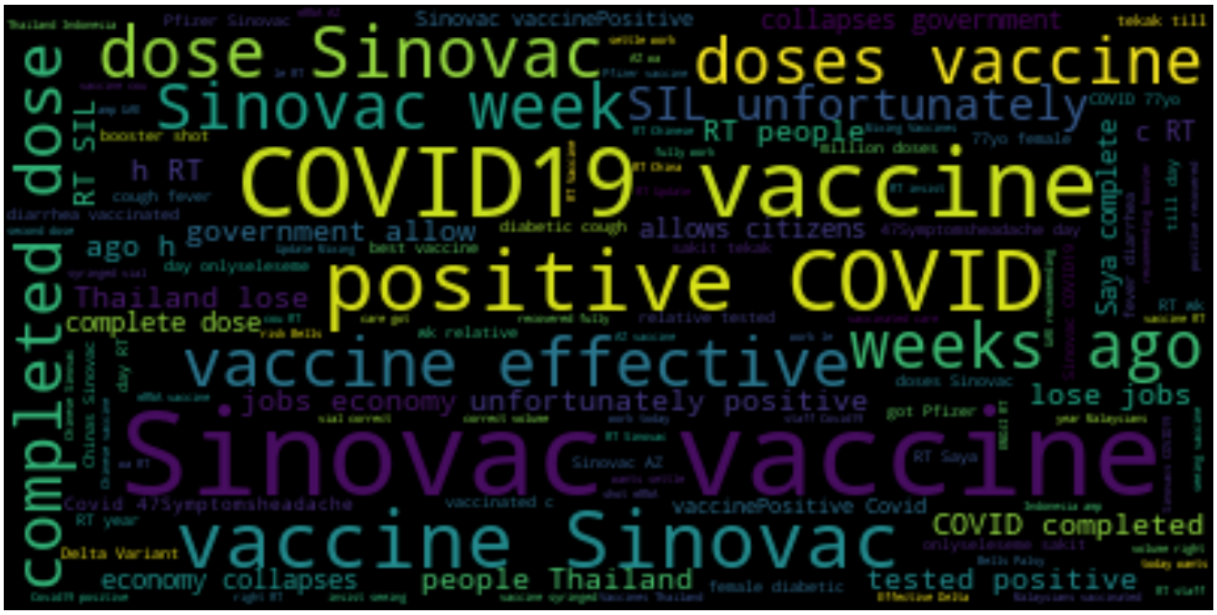
In [87]:
```
def plot_wordcloud(docx):
  plt.figure(figsize=(20,10))
  mywordcloud = WordCloud().generate(docx)
  plt.imshow(mywordcloud,interpolation='bilinear')
  plt.axis('off')
  plt.show()
```

In [88]:
```
pos_docx= ' '.join(pos_tokens)
```

In [89]:
```
neg_docx= ' '.join(neg_tokens)
```

In [90]:
```
neu_docx= ' '.join(neut_tokens)
```

In [91]:
```
plot_wordcloud(pos_docx)
```



In [79]:
```
plot_wordcloud(neg_docx)
```

In [80]:
```
plot_wordcloud(neu_docx)
```

In [ ]: