BioNTech-Pfizer_Vaccine_Tweets

February 10, 2021

[1]:

```
import time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from nltk.sentiment import SentimentIntensityAnalyzer
from transformers import pipeline
import warnings
warnings.filterwarnings('ignore')
```

1 Data

The data consists of tweets scraped from Twitter using Tweepy/TwitterApi. The 'pfizer_scrape.ipynb' file contains the scraping code. The data is a csv file, containing tweets relating to BioNTech/Pfizer COVID-19 vaccine.

The columns consist of the following: - username: username - acctdesc: acount description - location: location where the tweet originated from - following: number of accounts this user follows - followers: number of accounts that are following this user - totaltweets: total number of tweets from this user - usercreatedts: date of user account creation - tweetcreatedts: date of tweet - retweetcount: number of retweets this tweet has - favoritecount: number of favorites this tweet has - text: text of this specific tweet - hastags: hastags used in this tweet - id: tweet id

[2]:

```
df = pd.read_csv("vaccine_tweets.csv")
```

[3]:

```
df.head()
```

[3]:

```
          username                                           acctdesc  \
0        BrazilSFE  Brazil SFE®| We are passionate about improving…
1    _Indiaupdates  India Updates is an independent news & Informa…
2        TMReserve                         Join the real conversation
3  DrFariyaBukhari  Reality bites & so does my Blog. Dare to indul…
4   TheUltraAliens                                      Intuipreneur

            location  following  followers  totaltweets        usercreatedts  \
0  São Paulo, Brasil       1240         94        48337  2015-01-02 14:13:17
1   New Delhi, India        102        232        10937  2019-02-26 16:12:39
```

```
2           Malaysia       189       7352        73126  2011-05-05 16:27:46
3           Pakistan       254        597        53084  2014-04-20 14:54:05
4         Via Lactea      3141        722         7497  2014-11-01 08:39:00

        tweetcreatedts  retweetcount  favoritecount  \
0  2021-02-02 13:15:27             0              1
1  2021-02-02 13:15:00             0              0
2  2021-02-02 13:07:13             2              1
3  2021-02-02 13:05:28             1              1
4  2021-02-02 13:00:35             0              1

                                                text  \
0  Dê Like! https://t.co/wGCPT8qVpc\nGlobal Pharm…
1  Pfizer-BioNTech to produce 2 bn doses of Covid…
2  Pfizer forecasts $15b in Covid-19 vaccine sale…
3  Valid point. Only PCR negative &amp; Non-react…
4  "7 die at Spanish care home after getting #Pfi…

                                            hashtags                   id
0  [{'text': 'Top10', 'indices': [113, 119]}, {'t…  1356592067955339267
1  [{'text': 'Pfizervaccine', 'indices': [97, 111…  1356591952376987649
2  [{'text': 'Pfizer', 'indices': [59, 66]}, {'te…  1356589995620884481
3  [{'text': 'Pfizer', 'indices': [159, 166]}, {'…  1356589555126788096
4  [{'text': 'Pfizer', 'indices': [42, 49]}, {'te…  1356588324098572288
```

[4]:

```
def summary(x, columns=['text', 'retweetcount', 'favoritecount']):
    '''This function gives a summary of a datafame with the specified columns.
 ↪'''

   print('Results: {}'.format(len(x)))
   display(x[columns].head())
```

[5]:

```
summary(df)
```

```
Results: 2799

                                                text  retweetcount  \
0  Dê Like! https://t.co/wGCPT8qVpc\nGlobal Pharm...             0
1  Pfizer-BioNTech to produce 2 bn doses of Covid...             0
2  Pfizer forecasts $15b in Covid-19 vaccine sale...             2
3  Valid point. Only PCR negative &amp; Non-react...             1
4  "7 die at Spanish care home after getting #Pfi...             0

   favoritecount
0              1
1              0
2              1
3              1
```

```
4                   1
```

2 Sentiment Analysis

2.1 NLTK - VADER

NLTK has a sentiment analyzer called VADER(Valence Aware Dictionary and sEntiment Reasoner. The analyzer uses a lexical approach. Thismeans it uses words or vocabularies that have beenassigned predetermined scores. The VADER analyzer is broadly used on socialmedia. The accuracy is high on short text, butdecreases with larger texts.

Pros: * Fast * No pre-processing * Easy to use * Use of 3 classes * Short text accuracy

Cons: * Does not take context into account * Long text accuracy

[6]:

```
vader = SentimentIntensityAnalyzer()
```

[7]:

```
def get_sentiment_v(text):
    '''This function handles VADER results.'''
    if vader.polarity_scores(text)["compound"] > 0:
        return "POS"
    elif vader.polarity_scores(text)["compound"] < 0:
        return "NEG"
    else:
        return "NEU"
```

[8]:

```
vader_r = df.text.apply(get_sentiment_v)
```

Converting results into dataframe

[9]:

```
df['vader_label'] = vader_r
```

[10]:

```
summary(df, ['text', 'vader_label'])
```

```
Results: 2799

                                                text vader_label
0  Dê Like! https://t.co/wGCPT8qVpc\nGlobal Pharm...         POS
1  Pfizer-BioNTech to produce 2 bn doses of Covid...         NEU
2  Pfizer forecasts $15b in Covid-19 vaccine sale...         NEU
3  Valid point. Only PCR negative &amp; Non-react...         NEG
4  "7 die at Spanish care home after getting #Pfi...         POS
```

2.2 HuggingFace - Transformers

Transformers provides pre-trained models forNatural Language Understanding (NLU) andNatural Language Generation (NLG) tasks.Transformers has interoperability between Py-Torch and TensorFlow.

Pros: * No pre-processing * Easy to use * Takes context into account

Cons: * Relatively slow * Limited to 2 classes

```
[11]: transformers = pipeline('sentiment-analysis')
```

```
HBox(children=(FloatProgress(value=0.0, description='Downloading', max=230.0, style=ProgressSt
```

```
[12]: def get_sentiment_t(results):
          labels = []
          scores = []
          for result in results:
              result = result[0]
              labels.append(result['label'][:3])
              scores.append(result['score'])
          return labels, scores
```

```
[13]: start_run = time.time()
      transformers_r = df.text.apply(transformers)
      run_time = time.time() - start_run
      print('Run time: {} min {} sec'.format(round(run_time)/60, run_time%60))
      labels, scores = get_sentiment_t(transformers_r)
```

```
Run time: 1.1333333333333333 min 7.530554294586182 sec
```

Converting results into dataframe.

```
[14]: labels, scores = get_sentiment_t(transformers_r)

      df['transformer_label'] = labels
      df['transformer_score'] = scores
```

```
[15]: summary(df, ['text', 'vader_label', 'transformer_label', 'transformer_score'])
```

```
Results: 2799

                                                text vader_label  \
0  Dê Like! https://t.co/wGCPT8qVpc\nGlobal Pharm...         POS
1  Pfizer-BioNTech to produce 2 bn doses of Covid...         NEU
2  Pfizer forecasts $15b in Covid-19 vaccine sale...         NEU
3  Valid point. Only PCR negative &amp; Non-react...         NEG
4  "7 die at Spanish care home after getting #Pfi...         POS
```

```
  transformer_label  transformer_score
0               NEG           0.638096
1               NEG           0.992779
2               NEG           0.995393
3               NEG           0.968649
4               NEG           0.998614
```

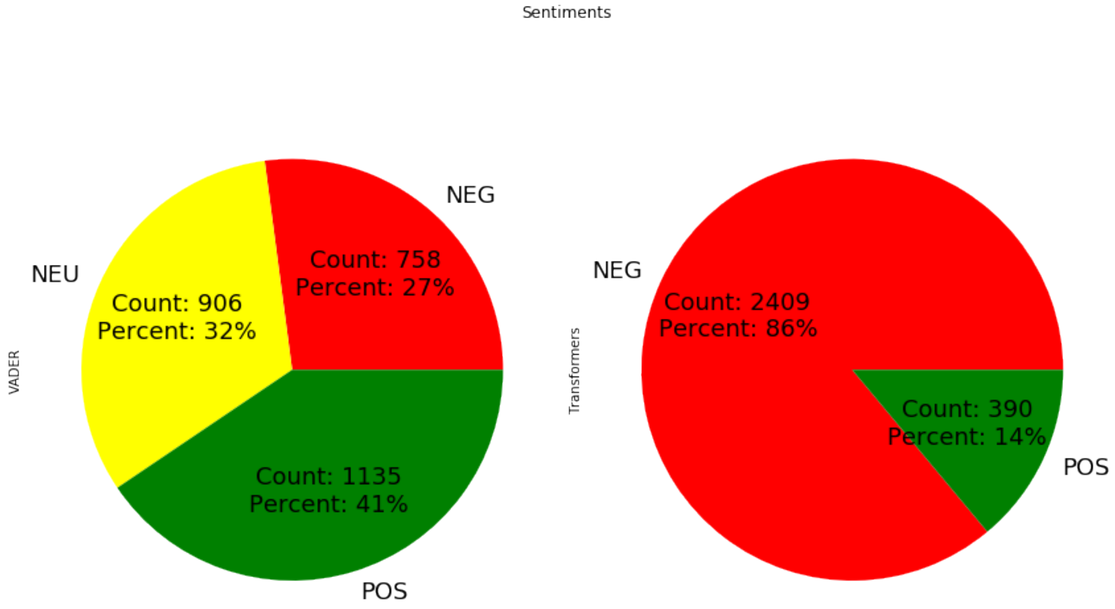
2.3 Results

[16]:

```
def get_label(x):
    return 'Count: {:.0f}\nPercent: {:.0f}%'.format(x/100*df_len, x)
```

[17]:

```
df_len = len(df)
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 8))
df_vader = df.groupby('vader_label').size()
df_transformer =df.groupby('transformer_label').size()
df_vader.plot(kind='pie', autopct=get_label, colors=['red', 'yellow', 'green'],␣
 ↪ax=ax1, label='VADER', textprops={'fontsize': 18})
df_transformer.plot(kind='pie', autopct=get_label, colors=['red', 'green'],␣
 ↪ax=ax2, label='Transformers', textprops={'fontsize': 18})
fig.suptitle('Sentiments')
plt.tight_layout()
plt.show()
```



Both models differ in results. VADER returns 3 classes whereas Transformers returns 2. This extra class is the neutral class. Because of this absence we remove the neutral results.

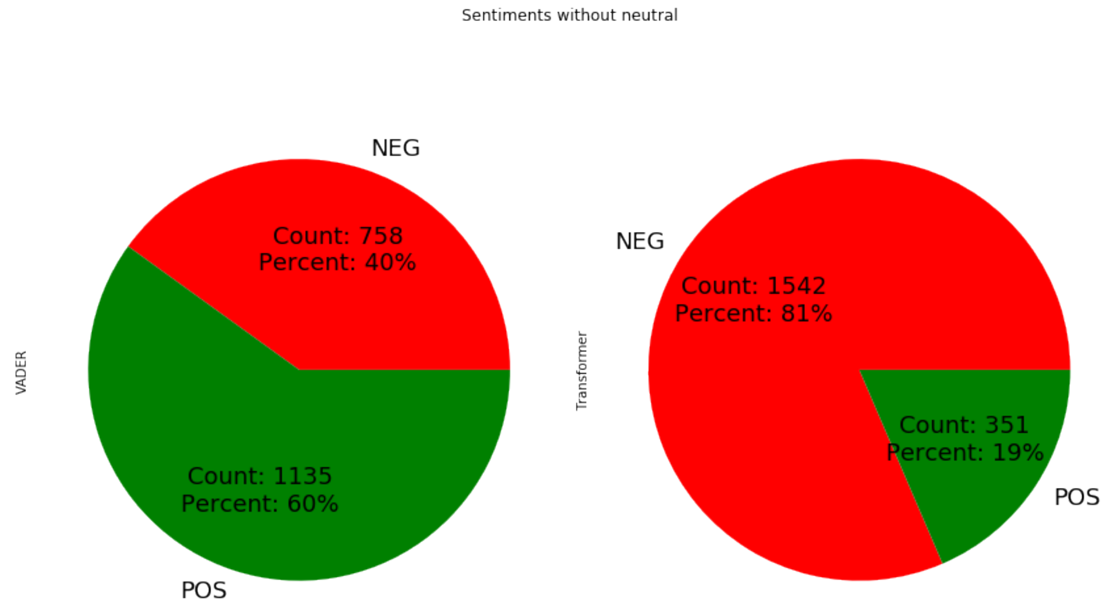
[18]:

```
df_posneg = df[df['vader_label'] != 'NEU']
df_len = len(df_posneg)
summary(df_posneg, ['text', 'vader_label', 'transformer_label',␣
 ↪'transformer_score'])
```

```
Results: 1893

                                                 text vader_label  \
0   Dê Like! https://t.co/wGCPT8qVpc\nGlobal Pharm...         POS
3   Valid point. Only PCR negative &amp; Non-react...         NEG
4   "7 die at Spanish care home after getting #Pfi...         POS
5   This is why we need to speed up the #vaccine p...         NEG
15  Well here you have it *the elderly are not pro...         POS

   transformer_label  transformer_score
0                NEG           0.638096
3                NEG           0.968649
4                NEG           0.998614
5                NEG           0.998797
15               NEG           0.998372
```

[19]:

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12, 8))
df_vader = df_posneg.groupby('vader_label').size()
df_transformer =df_posneg.groupby('transformer_label').size()
df_vader.plot(kind='pie', autopct=get_label, colors=['red', 'green'], ax=ax1,␣
 ↪label='VADER', textprops={'fontsize': 18})
df_transformer.plot(kind='pie', autopct=get_label, colors=['red', 'green'],␣
 ↪ax=ax2, label='Transformer', textprops={'fontsize': 18})
fig.suptitle('Sentiments without neutral')
plt.tight_layout()
plt.show()
```



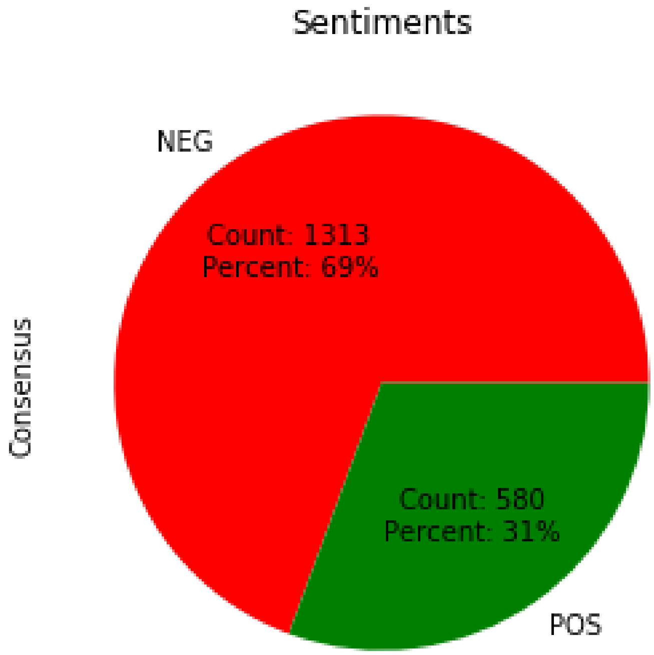
Both models give different results. The majority of VADER counts to positive, whereas Transformers tends to negative. Besides the differences, there are tweets at for which both models have the same result. Let's look at the consensus.

```
[20]: df_consensus =␣
       ↪df_posneg[df_posneg['vader_label']==df_posneg['transformer_label']]
      summary(df_consensus, ['text', 'vader_label', 'transformer_label',␣
       ↪'transformer_score'])
```

```
Results: 1051

                                                 text vader_label  \
3   Valid point. Only PCR negative &amp; Non-react...         NEG
5   This is why we need to speed up the #vaccine p...         NEG
19  Another company to check for #quarterly figure...         NEG
29  Anybody know anything about Covid #serology te...         NEG
30  'Given the significant impact that BNT162b2 is...         POS

   transformer_label  transformer_score
3                NEG           0.968649
5                NEG           0.998797
19               NEG           0.992188
29               NEG           0.998944
30               POS           0.868816
```

[21]:

```
df_cons_group = df_consensus.groupby('vader_label').size()
df_cons_group.plot(kind='pie', autopct=get_label, colors=['red', 'green'],␣
 ↪label='Consensus')
plt.title('Sentiments')
plt.tight_layout()
plt.show()
```



Engagement captures the amount of interactions a tweet has. For a tweet this will be views, favorites and retweets. We only have data for favorites and retweets, these will be used to define engagement.

$engagement = retweets + favorites$

[22]:

```
def get_engagement(x):
    return x.sum(axis=1)

engagement = get_engagement(df_consensus[['retweetcount', 'favoritecount']])
df_consensus['engagement'] = engagement
```

The transformers score goes for both labels from 0.5-1.0. Here, 1.0 is 100% positive or negative and 0.5 is 50% positive or negative. These scores are mapped onto a scale from 0.0-1.0. Now 0.0 is 100% negative and 1.0 is 100% positive. This score is captured into 'score_n'.

[23]:

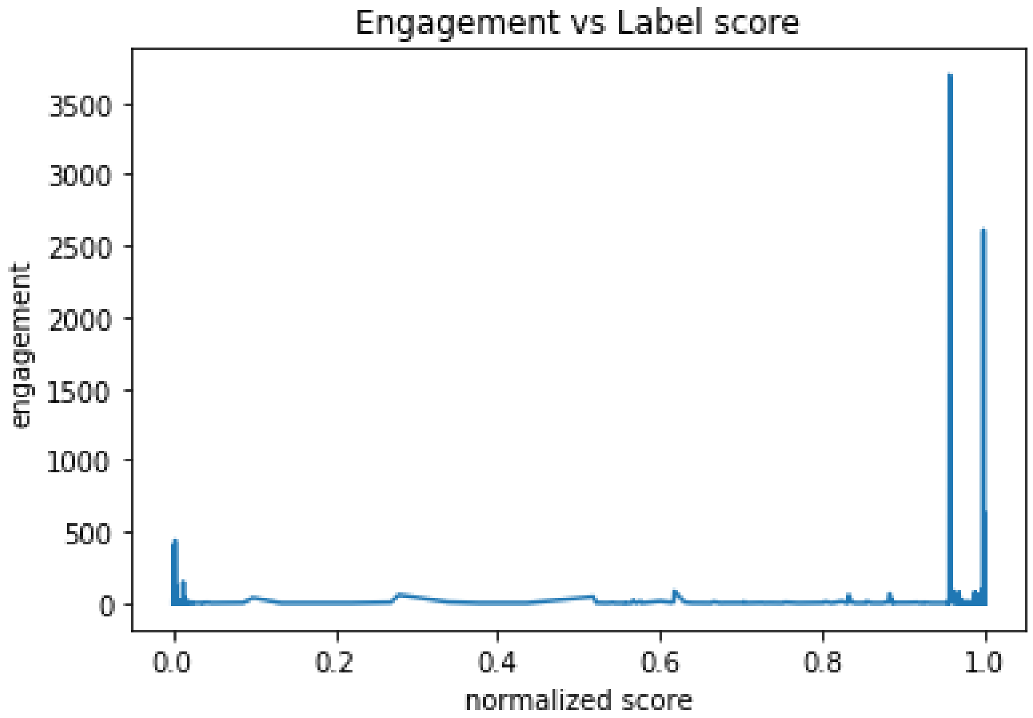
```
def get_normalize(x):
    if x[0] == 'NEG':
        return 1-float(x[1])
    else:
        return float(x[1])

df_consensus['score_n'] = df_consensus[['transformer_label',␣
 ↪'transformer_score']].apply(get_normalize, axis=1)
summary(df_consensus, ['text', 'vader_label', 'transformer_label',␣
 ↪'transformer_score', 'engagement', 'score_n'])
```

```
Results: 1051

                                                text vader_label  \
3   Valid point. Only PCR negative &amp; Non-react...         NEG
5   This is why we need to speed up the #vaccine p...         NEG
19  Another company to check for #quarterly figure...         NEG
29  Anybody know anything about Covid #serology te...         NEG
30  'Given the significant impact that BNT162b2 is...         POS

   transformer_label  transformer_score  engagement   score_n
3                NEG           0.968649           2  0.031351
5                NEG           0.998797          25  0.001203
19               NEG           0.992188           0  0.007812
29               NEG           0.998944           1  0.001056
30               POS           0.868816           0  0.868816
```

Now we can plot and see what tweets have the most engagement.

[24]:

```
df_engagement = df_consensus[['retweetcount','favoritecount', 'engagement',␣
 ↪'score_n']].sort_values(by=['score_n'])
plt.plot(df_engagement['score_n'], df_engagement['engagement'])
plt.title('Engagement vs Label score')
plt.ylabel('engagement')
plt.xlabel('normalized score')
plt.show()
```



[25]:
```
df_sort = df_consensus.sort_values(by=['engagement'], ascending=False)
ratio = df_sort.groupby('vader_label')['engagement'].sum()
total = ratio.sum()
n,p = ratio['NEG']/total, ratio['POS']/total
print('Ratio of total\nNEG: {}\nPOS: {}'.format(n,p))
```

```
Ratio of total
NEG: 0.31806976917905433
POS: 0.6819302308209457
```

It is clear to see that the most engagement is located at the extremes. However, it also shows that the positive tweets have more engagement. The positive side has a ratio of 0.68 against 0.32 of the negative side. Let's look at some examples of tweets with high engagement.

[26]:
```
def show_tweet(x):
    for _, tweet in x.iterrows():
        print('-------------------------------')
        print(tweet.text)
        print('Retweets: {} | Favorites: {}'.format(tweet.retweetcount, tweet.
 ↪favoritecount))
        print('-------------------------------')
```

Negative tweets

[27]:
```
show_tweet(df_sort[df_sort['vader_label'] == 'NEG'].head())
```

```
-------------------------------
Ontario has given 350K #CovidVaccines in 50 days, yet only vaccinated 61K/70K
#LTC residents-this population accounts for 2/3 of all Ontario #COVID19 deaths.

Prioritizing "speed over precision" and not moving #Pfizer vaccines into #LTC
homes until Jan 5th were deadly mistakes. https://t.co/AhjCGRnhot
Retweets: 124 | Favorites: 314
-------------------------------
-------------------------------
#Pfizer withdraws application for #COVID vaccine emergency use in India after
regulator requests more data

https://t.co/KsidlUPfAO https://t.co/yGTq6ZM6oM
Retweets: 161 | Favorites: 240
-------------------------------
-------------------------------
#Pfizer's second dose didn't seem to help

https://t.co/87SmXYRq3A
Retweets: 119 | Favorites: 162
-------------------------------
-------------------------------
@4dpharmaplc announces 2nd clinical collaboration for MRx0518 with a #checkpoint
inhibitor, with #Merck KGaA $MRK &amp; #Pfizer's $PFE #Bavencio (#avelumab) as
first-line maintenance therapy for #urothelial carcinoma, the most common form
of bladder cancer

https://t.co/I9EtUPaaBB
Retweets: 40 | Favorites: 109
-------------------------------
-------------------------------
#Pfizer withdraws application for emergency use of its COVID-19 vaccine in India
Retweets: 12 | Favorites: 119
-------------------------------
```

Positive tweets

[28]:
```
show_tweet(df_sort[df_sort['vader_label'] == 'POS'].head())
```

```
-------------------------------
We are working with pharmaceutical companies to ensure vaccines are delivered to
Europeans.

#BioNTech/@pfizer will deliver 75 million of additional doses in the second
quarter of the year - and up to 600 millions in total in 2021.
Retweets: 698 | Favorites: 2997
```

```
-------------------------------
-------------------------------
 I’ve got good news! David Lynch got his first dose of the #Pfizer vaccine.
https://t.co/DVHq4pMeDy
Retweets: 244 | Favorites: 2364
-------------------------------
-------------------------------
Update: my mom has been 48 hours since receiving the #Pfizer vaccine. She had a
little nausea; that’s it! Now feels perfect.
Retweets: 13 | Favorites: 621
-------------------------------
-------------------------------
It’s a BEAUTIFUL day to get Dose 2 of the vaccine.

I feel lucky, grateful, and hopeful. #Pfizer #CovidVaccine
https://t.co/GoZh1cgUH2
Retweets: 3 | Favorites: 103
-------------------------------
-------------------------------
Second dose of #PfizerBioNTech vaccine today. So unbelievably blessed and
fortunate. @CommHealthMW https://t.co/yTQHx4m3IL
Retweets: 4 | Favorites: 92
-------------------------------
```