

SENTIMENT ANALYSIS

A Case Study of Talabat Company

About Talabat

Talabat is an online food ordering company founded in Kuwait in 2004. As of April 2021, Talabat operates in Kuwait, Saudi Arabia, Bahrain, the United Arab Emirates, Oman, Qatar, Jordan, Egypt, and Iraq.

About The Dataset

Context

This dataset contains information scraped from twitter for the period June to November 2022. I used python script to scrape tweets with the keyword "Talabat". The tweets were saved to CSV in 3 progressive files. A total of 9,006 tweets were collected and processed.

This is what the original CSV dataset looks like after scraping and formatting into CSV row-column:

	A	B	C	D	E	F	G
1		User	Date	Text	Location	Likes	Source
2	0	Talabat	2022-10-31 13:03:46+00:00	@zchculdud Hello James We apologize for that, please check your DM	Dubai	0	Emplifi
3	1	Talabat	2022-10-31 12:47:47+00:00	@KVenturesMC Ù...ø±ø-ø`ø§ Ù±ø'ø±ø'ø± Ù...Ù±Ùf ø'Ù± ø§Ù§ ø§ø²ø'ø§ø- øµø§ø±,	Kuwait	10	Emplifi
4	2	Talabat	2022-10-31 12:22:31+00:00	@taibaalbaker Hello Taiba We apologize for the experience, please send us your		0	Emplifi
5	3	Talabat	2022-10-31 12:16:50+00:00	@MidhunrajRajanT Hello We apologize for that, please send us a DM with your order		0	Emplifi
6	4	lossing_	2022-10-31 12:13:18+00:00	@Talabat ÙfÙ,, ø§Ù,,Ù...ø-ø§ø'Ù... ø§Ù,,Ù§ ø§ø-Ù,,ø`Ù±ø§ Ù§ø±ø£ø*ø± ø§Ù,,ø-Ù,,ø`	Oman	0	Twitter for iPhone
7	5	g66nv	2022-10-31 11:56:10+00:00	@Talabat ø±Ù... ø-ø.ø± ø-ø²ø§ø`Ù§ Ù§Ù§ ø-Ù,,ø`ø§øø ø'Ù±Ù...ø§Ù± øY±:øY±² ø`ø²ø`ø` ø§Ù±Ù§ ø`ø*Ù,,ø±		0	Twitter for iPhone
8	6	Talabat	2022-10-31 11:44:00+00:00	@asm909r Ù...ø±ø-ø`ø§ ø'ø`ø`ø§Ù,,Ù,,Ù± Ù±ø'ø±ø'ø± Ù...Ù±Ùf ø'Ù± the horror of driving next to a talabat driver	øYžc	0	Emplifi
9	7	numnumalqassab	2022-10-31 11:43:19+00:00	@Talabat you are really testing my patience. Have me wait on an order that never came and		0	Twitter for iPhone
10	8	zchculdud	2022-10-31 11:27:51+00:00		Salalah, Om Ma	0	Twitter for iPhone

Objective Of The Analysis

The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a negative sentiment associated with it. So, the task is to classify negative, or positive tweets from other tweets based on how twitter users feel about Talabat as a company and it's services.

Data Analytic Tool

Python programming language was used in this project, putting into use the Natural language processing technique among other features.

The needed modules were imported and used. These includes:

```
##Import modules
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import string
import nltk
import warnings
from nltk.stem.porter import *
from wordcloud import WordCloud
```

Data Preparations & Processing

The 3 CSV files created from the twitter scrapping were named "talabat.csv", "talabat1.csv", and "talabat2.csv". The 3 files were imported into dataframe on python for processing to begin.

```
##Import dataset
data = pd.read_csv('talabat.csv')
data1 = pd.read_csv('talabat1.csv')
data2 = pd.read_csv('talabat2.csv')
```

Then, they were merged into a single dataset called "talabat_data" and then into a single csv file.

```
##Merge the datasets
talabat_temp = pd.concat((data,data1), axis=0)
```

```
talabat_data = pd.concat((talabat_temp,data2), axis=0)
```

```
talabat_data.to_csv('talabat_data.csv', encoding='UTF-8')
```

```
#Inspecting the dataset
```

```
talabat_data.head(30)
```

```
   User      Date ... Likes      Source
0  Talabat  2022-10-31 13:03:46+00:00 ... 0      Emplifi
1  Talabat  2022-10-31 12:47:47+00:00 ... 0      Emplifi
2  Talabat  2022-10-31 12:22:31+00:00 ... 0      Emplifi
3  Talabat  2022-10-31 12:16:50+00:00 ... 0      Emplifi
4  lossing_ 2022-10-31 12:13:18+00:00 ... 0  Twitter for iPhone
5  g66nv    2022-10-31 11:56:10+00:00 ... 0  Twitter for iPhone
6  Talabat  2022-10-31 11:44:00+00:00 ... 0      Emplifi
7  numnumalqassab 2022-10-31 11:43:19+00:00 ... 0  Twitter for iPhone
8  zchculdud 2022-10-31 11:27:51+00:00 ... 0  Twitter for iPhone
9  jordanfinance 2022-10-31 11:04:47+00:00 ... 0  Twitter for iPad
10 KVenturesMC 2022-10-31 09:26:46+00:00 ... 3  Twitter Web App
11 MohdHALMoosa 2022-10-31 08:46:50+00:00 ... 2  Twitter for Android
12 SimplyNadzz 2022-10-31 08:10:16+00:00 ... 0  Twitter for iPhone
13 tambi_jalouqa 2022-10-31 07:23:47+00:00 ... 0  Twitter for iPhone
14 taibaalbaker 2022-10-31 07:18:59+00:00 ... 0  Twitter for iPhone
15 riad_aj    2022-10-31 07:06:31+00:00 ... 0  Twitter Web App
16 Chris_Kaye001 2022-10-31 06:37:48+00:00 ... 15  Twitter Web App
17 shahadjkh 2022-10-31 06:23:11+00:00 ... 0  Twitter for iPhone
18 taibaalbaker 2022-10-31 05:49:15+00:00 ... 0  Twitter for iPhone
19 MidhunrajRajanT 2022-10-31 05:24:35+00:00 ... 0  Twitter for iPhone
20 tambi_jalouqa 2022-10-31 05:20:55+00:00 ... 3  Twitter for iPhone
21 riad_aj    2022-10-31 05:13:08+00:00 ... 16  Twitter for iPhone
22 guardiansofsea 2022-10-31 03:15:31+00:00 ... 1  Twitter for iPhone
23 Talabat  2022-10-31 00:56:05+00:00 ... 0      Emplifi
24 asm909r  2022-10-31 00:16:11+00:00 ... 0  Twitter for iPhone
25 Talabat  2022-10-30 23:52:17+00:00 ... 0      Emplifi
26 alyousif_fatma 2022-10-30 23:23:39+00:00 ... 1  Twitter for iPhone
27 Talabat  2022-10-30 22:57:17+00:00 ... 0      Emplifi
28 madollax 2022-10-30 22:52:27+00:00 ... 0  Twitter for iPhone
29 madollax 2022-10-30 22:38:02+00:00 ... 0  Twitter for iPhone

[30 rows x 6 columns]
```

Initial data cleaning requirements that we can think of after looking at the top records:

- The Twitter handles @user are hardly giving any information about the nature of the tweet.
- We also need to get rid of the punctuations, numbers and even special characters.
- Most of the smaller words do not add much value. For example, 'and', 'his', 'all'. So, we will try to remove them as well from our data.
- We can also split every tweet into individual words (tokenization), since it is an essential step in any NLP task.
- We need to shrink related words. We have terms like loves, loving, lovable, etc. in the rest of the data. These terms are often used in

the same context. If we can reduce them to their root word, which is 'love', then we can reduce the total number of unique words in our data without losing a significant amount of information.

```
## Function to remove pattern
def remove_pattern(input_text, pattern):
    r = re.findall(pattern, input_text)
    for i in r:
        input_text = re.sub(i, "", input_text)
    return input_text

# remove twitter handles
talabat_data['clean_tweet'] = np.vectorize(remove_pattern)(talabat_data['Text'],
"@[\w]*")

#remove special characters, numbers and punctuations
talabat_data['clean_tweet'] = talabat_data['clean_tweet'].str.replace("[^a-zA-Z#]", "")

#remove special characters, numbers and punctuations for location
talabat_data['clean_location'] = talabat_data['Location'].str.replace("[^a-zA-Z#]", "")
talabat_data['clean_location'] = talabat_data['clean_location'].str.strip()

#Remove stop words
talabat_data['clean_tweet'] = talabat_data['clean_tweet'].apply(lambda x: ' '.join([w
for w in x.split() if len(w)>3]))

#talabat_data.loc[:,['Date', 'Text', 'clean_tweet']].head()

```

Date	Text	clean_tweet
2022-10-31 13:03:46+00:00	@zchculdud Hello James\nWe apologize for that,...	Hello James apologize that please check your
2022-10-31 12:47:47+00:00	@KVenturesMC ôÉ±_¬_α \nô _»_î± ôÉô ôÄ _...	
2022-10-31 12:22:31+00:00	@taibalbaker Hello Taiba\nWe apologize for th...	Hello Taiba apologize experience please send y...
2022-10-31 12:16:50+00:00	@MidhunrajRajanT Hello \nWe apologize for that...	Hello apologize that please send with your ord...

2022-10-31
12:13:18+00:00

@Talabat
_ _ _ _ _
_ _ _ _ _

```
#Tokenize each tweet - splitting a string of texts into tokens/list
tokenize_tweets = talabat_data['clean_tweet'].apply(lambda x: x.split())
tokenize_tweets.head()
```

```
#Stemming or shrinking related words in the tokenized list
from nltk.stem.porter import *
stemmer = PorterStemmer()
```

```
tokenize_tweets = tokenize_tweets.apply(lambda x: [stemmer.stem(i) for i in x])
tokenize_tweets.head()
```

```
0 [Hello, James, apologize, that, please, check,...
1 []
2 [Hello, Taiba, apologize, experience, please, ...
3 [Hello, apologize, that, please, send, with, y...
4 []
```

Name: clean_tweet, dtype: object

```
#join the token back together
for i in range(len(tokenize_tweets)):
    tokenize_tweets[i] = ' '.join(tokenize_tweets[i])
```

```
talabat_data['clean_tweet'] = tokenize_tweets
```

```
#before
21 [Food, delivery, aggregators, like, should, ha...
22 [Thank, Kuwait, helping, clean, beaches, prote...
23 [Mohammed, Sorry, hear, about, this, have, fol...
```

```
#after
21 [food, deliveri, aggreg, like, should, have, o...
22 [thank, kuwait, help, clean, beach, protect, m...
23 [moham, sorri, hear, about, thi, have, follo
```

```
talabat_data.head(30)
```

```
      User ...          clean_tweet
0  Talabat ...  hello jame apolog that pleas check your
1  Talabat ...
2  Talabat ...  hello taiba apolog experi pleas send your orde...
3  Talabat ...  hello apolog that pleas send with your order n...
4  lossing_ ...
5  g66nv ...
6  Talabat ...
7  numnumalqassab ...  horror drive next talabat driver
8  zchculdud ...  realli test patienc have wait order that never...
9  jordanfinance ...  cloud kitchen already over what will happen th...
10 KVenturesMC ...          paci #usepaci http rabw
```

Visualization From Tweets Dataset

We will explore the cleaned tweets text. Exploring and visualizing data, no matter whether its text or any other data, is an essential step in gaining insights.

A few probable questions questions related to the data in hand are:

- What are the most common words in the entire dataset?
- Does the most common words in the dataset reflect negative or positive tweets about Talabat?
- What are the most used hashtags in these tweets and what do they reflect?

a. Let's discover the common words used in the tweets via WordCloud

A wordcloud is a visualization wherein the most frequent words appear in large size and the less frequent words appear in smaller sizes.

```
#word cloud generation
```

```
all_words = ' '.join([text for text in talabat_data['clean_tweet']])
```

```
from wordcloud import WordCloud
```

```
wc = WordCloud(width=750, height=450, random_state=21,  
max_font_size=110).generate(all_words)
```

```
plt.figure(figsize=(10,7))
```



```
# function to collect hashtags
def hashtag_extract(x):
    hashtags = []
    # Loop over the words in the tweet
    for i in x:
        ht = re.findall(r"#(\w+)", i)
        hashtags.append(ht)

    return hashtags

# extracting hashtags from all tweets
HT_all = hashtag_extract(talabat_data['clean_tweet'])

# unnesting list
HT_all = sum(HT_all,[])

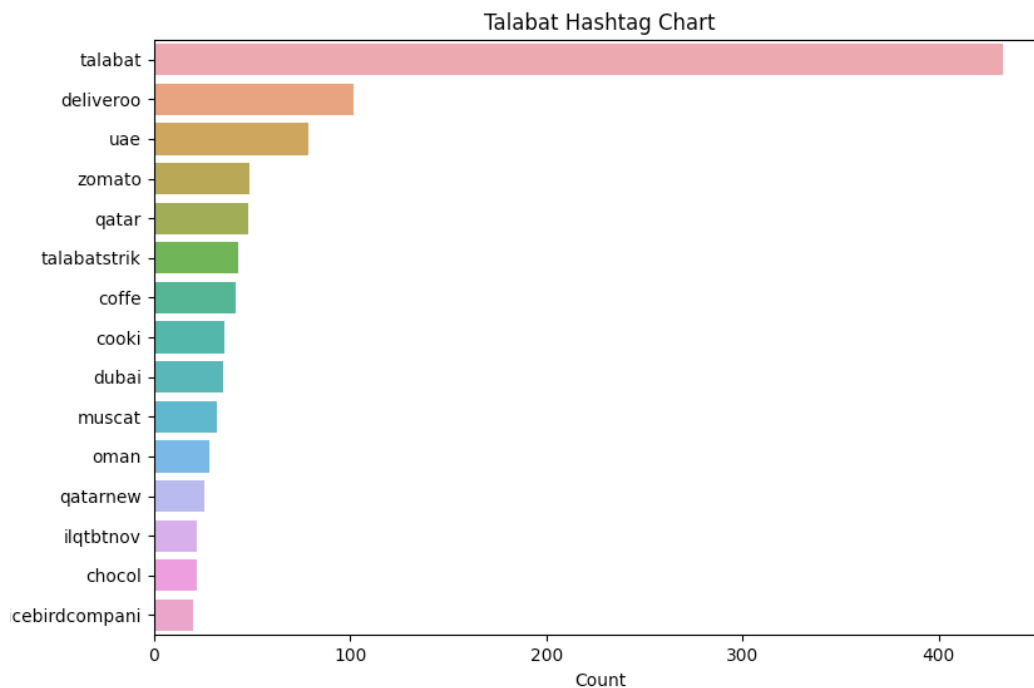
a = nltk.FreqDist(HT_all)
d = pd.DataFrame({'Hashtag': list(a.keys()),
                  'Count': list(a.values())})

# selecting top 15 most frequent hashtags
d = d.nlargest(columns="Count", n = 15)
plt.figure(figsize=(16,5))

ax = sns.barplot(data=d, x= "Count", y = "Hashtag", orientation='horizontal')

ax.set(title = 'Talabat Tweets Hashtags')

plt.show()
```

Observation:

Once again, the chart shows us that most hashtags used are neutral: location based and service based.

Finally, let us get the polarity from cleaned tweets

Sentiment polarity for an element defines the orientation of the expressed sentiment. It determines if the text expresses the positive, negative or neutral sentiment of the user about the entity in consideration, in our case, "TALABAT".

```
# Polarity score
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

```
# adding a row_id field to the dataframe, which will be useful for joining dataframes later
```

```
talabat_data["row_id"] = talabat_data.index + 1
```

```

#create a new data frame with "id" and "comment" fields
df_subset = talabat_data[['row_id', 'Text']].copy()

#covert to lower-case
df_subset['clean_tweet'] = df_subset['clean_tweet'].str.casefold()

# set up empty dataframe for staging output
df1=pd.DataFrame()
df1['row_id']='9999999999'
df1['sentiment_type']='NA999NA'
df1['sentiment_score']=0

print('Processing sentiment analysis...')
sid = SentimentIntensityAnalyzer()
t_df = df1
for index, row in df_subset.iterrows():
    scores = sid.polarity_scores(row[1])
    for key, value in scores.items():
        temp = [key,value,row[0]]
        df1['row_id']=row[0]
        df1['sentiment_type']=key
        df1['sentiment_score']=value
        t_df=t_df.append(df1)

#remove dummy row with row_id = 9999999999
t_df_cleaned = t_df[t_df.row_id != '9999999999']

#remove duplicates if any exist
t_df_cleaned = t_df_cleaned.drop_duplicates()

# only keep rows where sentiment_type = compound
t_df_cleaned = t_df[t_df.sentiment_type == 'compound']

df_output = talabat_data.merge(t_df_cleaned, on='row_id', how='inner')

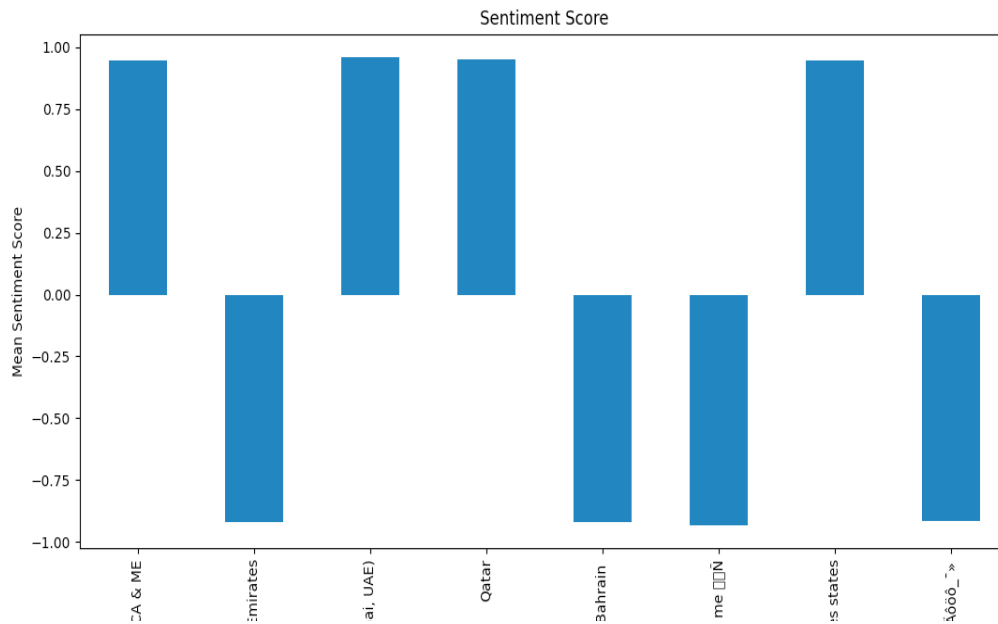
df_output[["sentiment_score"]].describe()
df_output1 = df_output['clean_location'].nlargest(10)

#generate mean of sentiment_score by period
dfg = df_output.groupby(['clean_location'])['sentiment_score'].mean()

#create a bar plot
plt.xlabel('Location')
plt.ylabel('Mean Sentiment Score')

```

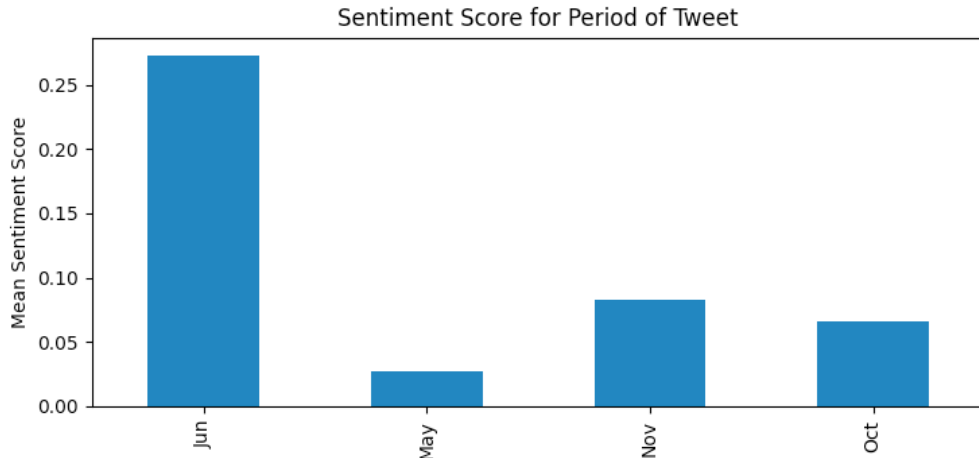
```
dfg.plot(kind='bar', title='Sentiment Score', y='sentiment_score', x='clean_location',
figsize=(6, 5))
plt.show()
```



```
df_output_mnt = df_output
```

```
#### show for date-month grouping
#generate mean of sentiment_score by period
dfg = df_output_mnt.groupby(['mnt'])['sentiment_score'].mean()
```

```
#create a bar plot
plt.xlabel('Period of Tweet')
plt.ylabel('Mean Sentiment Score')
dfg.plot(kind='bar', title='Sentiment Score', y='sentiment_score', x='Location',
figsize=(6, 5))
plt.show()
```



Conclusion

Generally, Talabat company is perceived by most twitter users sampled as a good company to deal with, and interestingly, there are no negative tweets across the different countries in which Talabat operates.