Sentiment Analysis of Twitter Data on Online Food Ordering Services using Python

Pragya Alagh
Department of computer Science
Indraprastha College For Women,
University Of Delhi
Email: pralagh@gmail.coml

Huda
Department of computer Science
Indraprastha College For Women,
University Of Delhi
Email: hudarayeen@gmail.com

Diksha Jain
Department of computer Science
Indraprastha College For Women,
University Of Delhi
Email:dikshajain14@gmail.com

***Abstract*—We live in a world where the amount of data that is being generated online is increasing every day. Various social media platforms can be used to collate data about any conceivable business, trend, political movement and target group to generate valuable insights. Natural Language Processing is one such tool that is gaining traction to understand the kind of interactions that happen on the internet and to optimize consumer experiences with the use of text data. By combining Python's capability to process the sentiments in text data and its ability to plot and analyse data in meaningful ways, this paper aims to develop a stepwise approach in which twitter data can be analysed in a way that can make a real business impact. For the purpose of this paper, we will analyse data retrieved from Twitter on the few popular online food ordering services in India and attempt to generate graphs that will be valuable in making recommendations to enhance business practices.**

I. Introduction

Sentiment analysis (SA) is a technique to analyze the emotions depicted through written text.

The internet is a huge repository for communication between the customers and service providers. Social networking sites are emerging platforms where people express themselves and write about their daily experiences. In recent times social media has become a platform for advertising and fostering customer relationshipss. Along with other things people also use it as a platform to talk about their experiences with various products and businesses. This in fact can be utilized to generate an insight of how people feel about, a certain product or service. The idea is to leverage all the available data and then analyze the extracted data in order to understand how customers engage with businesses, and to measure the effectiveness of various advertising campaigns. Through this paper, we wish to explore the data available on the social media platform twitter to analyze customer relationships with online food ordering services.

There is a bunch of evolving technologies that are being used for sentiment analysis. Vishal A. Kharde and S.S. Sanoawane in their paper on sentiment analysis [1] of twitter data divided the various approaches for sentiment analyses into two main categories. Machine Learning approaches and Lexicon based approaches. We will be using VADER(Valence Aware Dictionary for Sentiment Analysis) that is built on lexical approach. The task that we will be performing on the data can be listed down in following steps.

1) Data Scrapping from Twitter using tweepy
2) Cleaning data using pandas and numpy.
3) Analysis of data using NLTK and VADER
4) Using matplotlib to generate graphs.
    a) Comparison based graphs and competitor analysis.
    b) Area Wise analysis

II. Sentiment Analysis

As stated in the paper on SA by Walaa Medhat, Ahmed Hassan and Hoda Korashy, [2] SA is the computational treatment of opinions, sentiments and subjectivity of text.

A. Applications of Sentiment Analysis

Sentiment Analysis algorithms can be used to solve a wide range of problems. For example, measuring similarity between two words by comparing their contextual distribution. This can be useful in classifying sentiments of articles. Yu and Wu [3] used a similar concept to analyze news articles. Their results showed that their method can discover more useful emotion words, and their corresponding intensity improves their classification performance. Many similar tasks can be performed from widely classified SA algorithm.

In political field, it is used to keep track of political views, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!

Another application of SA, also the one we will be concerned with in this paper is that, big companies and brands often use sentiment analysis for social media monitoring. There is a lot of data online on social networking sites such as Facebook and twitter in the form of tweets, status updates, blog posts etc that can be utilized to know how people actually feel regarding a service or product that a company offers.

B. Why we choose Twitter

Twitter is a social media platform that has been there since early twenties, right from the time when people were learning to become accustomed with social media culture. This is one of the major reason for the rise in popularity of Twitter. It offers its user a platform to express their views and thoughts about things that matter to them and allow them to build relationships with others who have similar interests. Apart from being a place for connecting with people social media has also become a big platform for businesses to advertise their products and services to buid a customer base and because of this people are often tempted to post about their experiences with businesses. Because of the user interface and interaction offered by Twitter it has become one of the top most site where people write about their experiences with some product they recently used or some service they recently availed. This lead us to choosing Twitter for our analysis. Python offers a range of libraries that can easily retreive data in form of JSON(Java Script Object Notation) from Twitter. After retreival the collected data can be examined for analyzing the sentiment that they display

III. Python technology in Sentiment Analysis

A. Existing Libraries

Below is the list of [4] top few python libraries on the basis of the number of stars they have bagged on GitHub till Feb 2018. Below we disscuss five of mostly liked and used libraries of python in brief.

1) spaCy: spaCy is an extremly optimized python library that is used together with deep learning frameworks such as TensorFlow and PyTorch. SpaCy exels at large scale information extraction tasks such as processing entire web dumps. With the help of staCy linguistically sophisticated statistical models can be constructed for a variety of natural language processing problems with great ease.

2) Gensim: Gensim is specially optimized for semantic modelling. It is known to be fast scaleble and very efficient in its specialized tasks that consist of semantic analysis and topic modelling. Other similar areas in which Gensim is used are:

- Analysing plain text document for semantic structure
- Retreiveing semantically similar documents

3) Pattern: Pattern is mainly a web mining module. Along with that it tackels natural language processing tasks including sentiment analysis. Apart from sentiment analysis pattern is useful for performing

- tagger/chunker
- n-gram search

Pattern can be used to crawl sites such as Google, Twitter, Wikipedia etc. and parse the collected data.

4) TextBLob: Similar to pattern, TextBlob is also a web mining tool popular with sentiment analysis researchers. It provides a simple API for diving into common natural language processing tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

5) NLTK (Natural Language Toolkit): As the name suggests NLTK is a tool box for solving problems related to natural language processing. It offers easy to use resourses for performing complicated tasks such as tokenization and semantic reasoning. Some of the major tasks for which NLTK is used are:

- Tokenize and tag text
- Identify names entities
- Display a parse tree

B. The approach we are using for sentiment analysis

Amongst all the libraries NLTK is the most robust and provides an extensive range of lexicons that can be used for sentiment analysis. For the purpose of this paper we will be using the VADER (Valence Aware Dictionary for sentiment Analysis) lexical approach support for which is provided by NLTK. We chose the VADER lexical approach because it has largely been developed for social media microblogging like content and hence, is in sync with the type of data that we are using for this paper. Additionally, VADER fares better in accuracy than most of the highly reputed lexical and machine learning based models that are used in the industry and in some cases outperforms even individual human raters. [25]

IV. Our Experiment

A. Why did we choose Online Ordering Services

The data available on Twitter is an extremely valuable resource to gather information about the interaction of any company with its customers. People often post about their experiences with various products and services that they use, with the online community. Lately, this has been especially prevalent on Twitter. This makes data collected from Twitter a great place to do market analysis, compare a business with its competitors and gather information about the demographic of the users of the service. This data can be extremely useful in making various business decisions. One of the segments for which this data can prove to be immensely useful is online food ordering services. These services often cater to their customer grievances through their Twitter handle, which means that they generate a huge amount of data about their customer interactions, service problem areas as well as how they fare against their competitors. Through this data set, we will demonstrate the methodology of analysing twitter data right from mining the data via python to creating visualizations that will help decision makers in understanding their businesses.

B. Methodology

The methodology employed in order to generate insights from the given data and to perform sentiment analysis is divided into three segments:-

1) Data Scraping and Mining using Python
2) Understanding the attributes of Twitter data
3) Arranging Data and Extracting useful information.

1) Data Scraping and Mining using Python: There are several libraries that are available in Python to mine data. The most prominent and extensive of them being Tweepy. We chose Tweepy amongst the alternatives because of it's excellent and huge developer and user [17]. It is regularly maintained in accordance to the new versions of Python. Tweepy provides support for both the REST API for Standard Search in Twitter and real-time data streaming from Twitter. There are certain limitations to the Standard Search such as:

- It can only retrieve data via a keyword search for the last 7 days.
- The data retrieved is not the full set of available tweets for the keyword but only the top searches. [21]

Due to these limitations we decided to Stream data in real-time for 9 days in order to get a more richer and accurate data set for our analysis. The data was streamed via Tweepy by applying filters [22] to capture the tweets that are relevant for online food ordering services. The following code was developed and used.

```
#Streaming the relevant Tweets from Twitter using
the Twitter API for Python called Tweepy.
#We are using the Streaming method of
Tweepy to Stream tweets in real time from Twitter.

class MyStreamListener(tweepy.StreamListener):

  def on_connect(self):
    print("We are connected to the Twitter server.")
    pass

  def on_data(self, raw_data):
    print(raw_data)
    return True

myStreamListener = MyStreamListener()
myStream = tweepy.Stream(auth = api.auth,
listener=myStreamListener,tweet_mode='extended',retry_time_start=
5.0)
myStream.filter(track=['swiggy','zomato','foodpanda'])

#the Filter method is used in order to filter the
relevant tweets by a set of keywords so that our
stream only picks up the tweets that have these keywords.
```

Fig. 1. Code for streaming live tweets from twitter

Post extraction, the data was saved in text files from where we collated all the data into a single Python Array called tweets_data.

```
#Code to import the data from the text file and then load
it into an Array in Python.
#We use the json library because the data retrieved
from Tweepy API is in the Java Script Object Notation
Format.

import json
import os
tweets_data=[]

def dataclean(path,tweetarray):
   open('temp.txt','w')
   for line in open(path):
      if not line.isspace():
open('temp.txt','a').write(line)
      else:
         continue
   for line in open('temp.txt', 'r'):
      try:
         tweet=json.loads(line)
         tweetarray.apppend(tweet)
      except:
         continue
   os.remove('temp.txt')
```

Fig. 2. Code for Cleaning Twitter Data into a useable format

After the data is cleaned and added to tweets_data, the 'text' attribute of each tweet is used to classify tweets according to the food ordering service that the tweet refers to. Through this classification we see that the tweets in our data set are divided in the following manner.

Online Food-Ordering Service	No. of Tweets retrieved
Zomato	6040
Swiggy	1570
Food Panda	1866
Unclassified	632
Total*	10,108

*There may be some approximation in the number of tweets for each individual service due to overlapping of tweets as users tagged two or more services in one tweet. It has been ignored because the overlaps are in the range <100.

Fig. 3. Number of tweets for each service

2) Understanding attributes of Twitter data: The first step towards extracting useful information from the data set that has been collated is to understand the various attributes of the data set. In order to determine information about the attributes present in our data set as well as the attributes that can be useful for our analysis we mapped the raw twitter data into a Pandas DataFrame object.

```
#Mapping the tweets_data[ ] array into a
DataFrame using Pandas

import pandas as pd
tweets=pd.DataFrame(tweets_data)
```

Fig. 4. Mapping Tweets data into a Pandas DataFrame

Post mapping our data into Pandas DataFrame, we can use the info() method of DataFrame object to see the attributes of the data and the amount of data that is available to us.

<class 'pandas.core.frame.DataFrame'>	
RangeIndex: 10108 entries, 0 to 10107	
Data columns (total 36 columns):	
contributors	0 non-null object
coordinates	41 non-null object
created_at	10108 non-null object
display_text_range	4105 non-null object
entities	10108 non-null object
extended_entities	651 non-null object
extended_tweet	4018 non-null object
favorite_count	10108 non-null int64
favorited	10108 non-null bool
filter_level	10108 non-null object
geo	41 non-null object
id	10108 non-null int64
id_str	10108 non-null object
in_reply_to_screen_name	4466 non-null object
in_reply_to_status_id	3000 non-null float64
in_reply_to_status_id_st	3000 non-null object
in_reply_to_user_id	4466 non-null float64
in_reply_to_user_id_str	4466 non-null object
is_quote_status	10108 non-null bool
lang	10108 non-null object
place	337 non-null object

Fig. 5.

place	337 non-null object
possibly_sensitive	3285 non-null object
quote_count	10108 non-null int64
quoted_status	911 non-null object
quoted_status_id	911 non-null float64
quoted_status_id_str	911 non-null object
quoted_status_permalink	911 non-null object
reply_count	10108 non-null int64
retweet_count	10108 non-null int64
retweeted	10108 non-null bool
retweeted_status	2490 non-null object
source	10108 non-null object
text	10108 non-null object
timestamp_ms	10108 non-null object
truncated	10108 non-null bool
user	10108 non-null object
dtypes: bool(4), float64(3), int64(5), object(	
memory usage: 2.5+ MB	

Fig. 6. Attributes of Twitter Data

From the above table we can see that we have a huge amount of data available for users, their coordinates, creating time for the tweets, extended entities like hashtags etc. Now that we understand the kind of attributes that are broadly available to us we can further segment them into more Data Frames that exclusively contain the information of users and the extended entities of the available tweets.

Upon further segmenting the data and mapping the sub-attributes of the given attributes we find that the data set also contains more information about uses and their location.

We create two additional Data Frames:-
i. The user_df that contains the data about all the users
ii. The extendedtweets_df that contains data about the extended entities of the tweet such as hashtags and the full_text attribute in case the tweet is more than 140 characters.

<class 'pandas.core.frame.DataFrame'>	
RangeIndex: 10108 entries, 0 to 10107	
Data columns (total 39 columns):	
contributors_enabled	10108 non-null bool
created_at	10108 non-null object
default_profile	10108 non-null bool
default_profile_image	10108 non-null bool
description	7139 non-null object
favourites_count	10108 non-null int64
follow_request_sent	0 non-null object
followers_count	10108 non-null int64
following	0 non-null object
friends_count	10108 non-null int64
geo_enabled	10108 non-null bool
id	10108 non-null int64
id_str	10108 non-null object
is_translator	10108 non-null bool
lang	10108 non-null object
listed_count	10108 non-null int64
location	7296 non-null object
name	10108 non-null object
notifications	0 non-null object
profile_background_color	10108 non-null object
profile_background_image_url	10108 non-null object
profile_background_image_url_https	10108 non-null object

Fig. 7.

profile_link_color	10108 non-null object
profile_sidebar_border_color	10108 non-null object
profile_sidebar_fill_color	10108 non-null object
profile_text_color	10108 non-null object
profile_use_background_image	10108 non-null bool
protected	10108 non-null bool
screen_name	10108 non-null object
statuses_count	10108 non-null int64
time_zone	0 non-null object
translator_type	10108 non-null object
url	3364 non-null object
utc_offset	0 non-null object
verified	10108 non-null bool
dtypes: bool(9), int64(6), object(24)	
memory usage: 2.4+ MB	

Fig. 8. Attributes of User Data

From the attributes of the user_df DataFrame we can see that the location attribute for users has 7296 non-null entities which means that we have location data available for a significant amount of the users who made tweets about online food ordering services. This data can be used to see the location of our user base and assess our audience.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4018 entries, 0 to 4017
Data columns (total 4 columns):
display_text_range     4018 non-null object
entities               4018 non-null object
extended_entities      1089 non-null object
full_text              4018 non-null object
dtypes: object(4)
memory usage: 125.6+ KB
```

Fig. 9. Extended Tweets Attribute

Now that we have full-information about the various attributes of the data, the amount of data that is available for each of these attributes we can move onto arranging this data in a meaningful way such that analysis can be performed.

3) Arranging data and extracting useful information: So far, we have a general overview of the various attributes of the data. We now need to assess which of these attributes can be of use to us and have a sufficient amount of sample space in order for us to perform some effective analysis.

1) Assessment of the Twitter engagement of online food ordering services in various cities.
2) Comparative analysis of the social media outlook of various online food ordering services in terms of positive and negative engagement.

For answering the aforementioned questions we need to extract and arrange and prepare the following kinds of data respectively:

1) Location of the users:
As we saw in Figure 6 the location of about 7296 users is available to us in the user_df DataFrame however, the total number of entities in this DataFrame is 10,108. Hence, we have to first extract the information of the users whose location is not null [23]

```
#DataFrame user_df enteries for which the
#location is not null.
frame=pd.DataFrame
frame=user_df[user_df.location.notnull()]
```

```
frame
```

10103	False	Tue Apr 28 12:55:28 +0000 2009	False
10104	False	Tue Feb 16 07:54:41 +0000 2016	True
10106	False	Tue May 08 13:53:20 +0000 2018	True
10107	False	Sun Jul 12 11:29:20 +0000 2009	False

7296 rows × 39 columns

Fig. 10. Data of users whose location is enabled

Once we retreive the user whose location is available we can easily group the location by city and extract the counts of unique users that belong to each city.

```
location_counts=frame.location.value_counts()
```

```
location_counts[:10]
```

```
India        1200
Mumbai        937
Delhi         737
Bangalore     572
Hyderabad     261
Kolkata       169
Pune          168
Chennai       159
Gurgaon       123
Noida          86
Name: location, dtype: int64
```

Fig. 11. Counting the number of tweets retrieved from each location

The above location counts are now in a format where they can be plotted into a graph.With the availability of this location data we can assess the city wise engagement of customers with online food ordering services. The next step is to clean and arrange text data in order to perform sentiment analysis.

2) Text data for Sentiment Analysis:
The text of the tweets extracted into an array called tweets_text[]. From the tweets_text[] we further make three arrays to classify tweets into tweets that are for Swiggy, Zomato and FoodPanda into three arrays swiggy _tweets[], zomato_tweets[] and panda_tweets[] respectively. These arrays are then used to perform sentiment analysis.

```
#extracting the text of the tweets from the
#entire data of the tweets to perform sentiment analysis.

count=0
tweets_text=[]
for tweet in tweets_data:
    if 'extended_tweet' in tweet:
        tweets_text.append
        (tweet['extended_tweet']['full_text'])
        count=count+1
    else:
        tweets_text.append(tweet['text'])
        count=count+1
print(count)
```

Fig. 12. Preparing tweets for sentiment analysis

From Figure 3 we can observe the distribution of the collected tweets over the three online food ordering services that we are concerned with.
Once the text data has been collated into separate arrays, we need extract the sentiment of each an every tweet grouped by the service that it is concerned with. For doing this we use the vader lexicon from the NLTK library. VADER's method polarity_scores() is used in order to determine the sentiment polarity of the text.

It assigns four values to each text string which are:-

a) 'pos' for positive, this score reflects the positive sentiment that is captured in the text string, it is in the range of 0.0 to 1.0 where 0.0 reflects the least positive affinity and 1.0 reflects the most positive affinity.
b) 'neg' for negative, this score reflects the negative sentiment that is captured in the text string, it is also in the range of 0.0 and 1.0 where 0.0 reflects the least negative affinity and 1.0 reflects the most negative affinity.
c) 'neu' for neutral, this score reflects the neutrality of the text string, the value for this score will be higher if the text is objective and doesn't have any sentiment attached to it and the value is lesser for texts that reflect strong positive or negative emotions.
d) 'Compound' for overall polarity score, the com-

pound score measures the overall negative or positive sentiment that the text depicts. The compound score is in the range of -1.0 to 1.0. A lower compound score signifies a more negative sentiment and a higher compound score reflects that the text string depicts strong positive emotions.

The compound score is calculated as the average of the positive, negative and neutral sentiment scores.

```
text="I have been very disappointed by \\
the behaviour of the staff \\
I would like to file a complaint immediately"
m=se.polarity_scores(text)
print(m)
```

```
{'neg': 0.253, 'neu': 0.634, 'pos': 0.113, 'compound': -0.4754}
```

Fig. 13. Generating polarity scores of text using VADER

In the above example (Fig. 13) we can see that the text depicts a fairly negative emotion and used words like ‘complaint’ and ‘disappointed’ hence, the neg value of the text is high as compared to the pos value and as a result the compound value is fairly negative which shows that the overall sentiment of the text is negative.

```
#function takes an array of tweets, assigns polarity
#scores to each tweet
#It maps the tweets and the polarity score and
#tweets in a DataFrame and then returns the DataFrame
#it marks the DataFrame entry with the company that
#it is concerned with
#by using the company_name parameter
def sentiment_scores_df(tweetsarray,company_name):
    senti={}
    for tweet in tweetsarray:
        m=se.polarity_scores(tweet)
        senti[tweet]=m
    df=pd.DataFrame.from_dict(senti, orient='index',
            columns=['neg','neu','pos','compound'])
    df['company']=company_name
    #data_frame.append(df,sort='false')
    return df

frame1=sentiment_scores_df(swiggy_tweets, 'swiggy')
frame2=sentiment_scores_df(zomato_tweets,'zomato')
frame2=sentiment_scores_df(zomato_tweets,'zomato')

#we append all the three DataFrame into a single frame
#to aid the process of making graphs
frame=frame1.append(frame2)
frame=frame.append(frame3)
```

Fig. 14. Code for generating polarity scores of individual tweets

	neg	neu	pos	compound	company
aan? Vantaanunga aatikittu.	0.000	1.000	0.000	0.0000	swiggy
rd plastic which definitely is on your part. @KTRTRS sir ryani https://t.co/aiEtH01jxR	0.186	0.749	0.065	-0.6096	swiggy
" superb timing annnooooo https://t.co/rywazmJ7Sv	0.000	0.760	0.240	0.6249	swiggy
" superb timing annnooooo https://t.co/5H0n4a01NU	0.000	0.760	0.240	0.6249	swiggy
n time #more than one hour	0.261	0.739	0.000	-0.6478	swiggy
rn a discount. Download on https://t.co/veLiSqBki5	0.000	1.000	0.000	0.0000	swiggy
https://t.co/7DZHELWM6k	0.000	1.000	0.000	0.0000	swiggy
cgj https://t.co/GDdelJCF8N	0.000	1.000	0.000	0.0000	swiggy

Fig. 15. Data Frame with tweets and corresponding polarity scores

V. Analysing Data and Generating Graphs

Post processing the data and mapping it to Data Frames such that the meaningful attributes of data are clearly visible and easily accessible, the next step is to reach certain conclusions from our data about the concerned online food ordering services. The best way to do this is to plot graphs that are easy to understand by the layman and can provide a visual appeal to the conclusions of the data that will be otherwise tough to assess.

We’ll plot the graphs using the matplotlib python library that has excellent integration with the pandas framework and can easily plot graphs from data frames.

We will plot the graphs for the following business assessments:-

1) Assessment of the Twitter engagement of online food ordering services in various cities.
2) Comparative analysis of the social media outlook of various online food ordering services in terms of positive and negative engagement.

A. Assessment of Twitter engagement

City wise social media engagement is an important metric for businesses to identify the states in which targeted social media campaigns can be run to get the maximum output and audience engagement. From the given data, after finding the top 10 cities that engage the most with the social media handles of various online food ordering services (Figure 3.2) we plot a bar graph that shows the engagement from each city and helps us easily recognize the cities with maximum engagement.

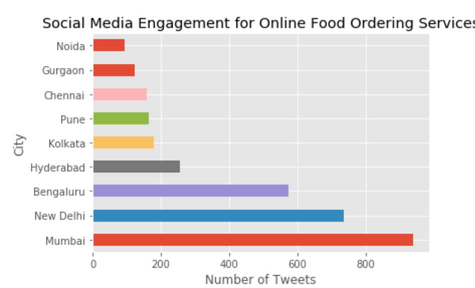

Fig. 16.

This shows us clearly that most number of tweets were retrieved from Mumbai and it has the highest engagement with online food ordering services on social media.

B. Comparative Analysis of Social Media Outlook

The main motive of this research was to conclusively determine which online food ordering service is the most popular amongst people and has the most positive outlook amongst users based on the reviews and complaints posted by users on Twitter. After performing sentiment analysis on individual tweets we collated the results and found out the mean positive and negative sentiments towards each service. According to the mean, we plotted the data on bar graphs for all three categories: positive, negative and overall sentiment towards the service

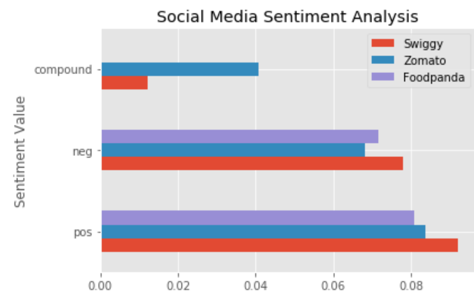

Fig. 17.

From the above graph we can see that the service with the most negative customer tweets is Swiggy and the service with the most positive customer tweets is also Swiggy. In this case a better metric to assess the performance of the services would be by looking at the compounded sentiment value. Since the compounded sentiment value is highest for zomato, we can safely say that in terms of social media outlook, the social media image and customer perception of Zomato is the most positive amongst the three and is the most negative for FoodPanda.

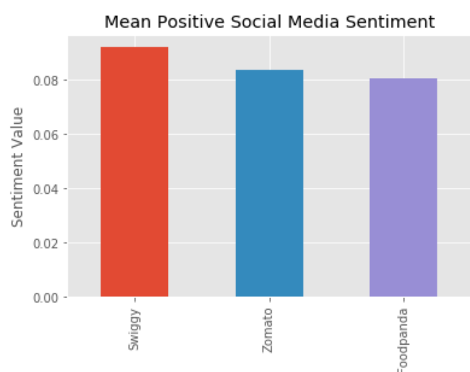

Fig. 18.

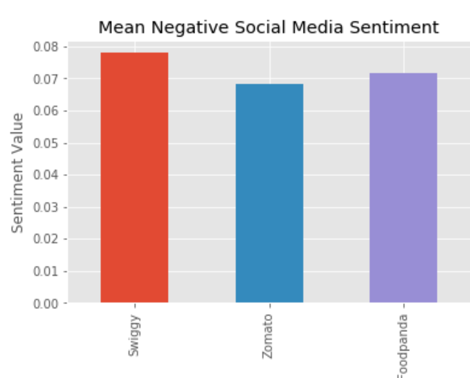

Fig. 19.

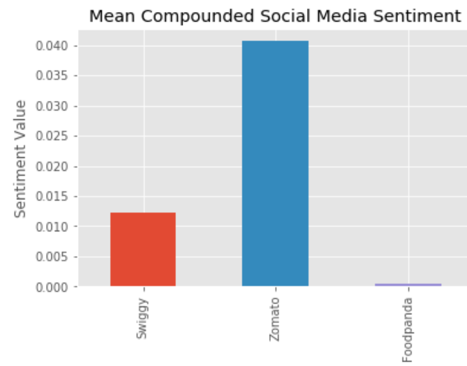

Fig. 20.

VI. Conclusion

Through this paper, we wanted to depict the methodology of generating insights from data scrapped from twitter. While, numerous authors and bloggers have made numerous posts about data analysis using twitter only a very few of them provide a detailed methodology which guides the reader through the whole process; from collection of data to creating the final graphs. This paper takes the reader through the entire process in minute details and using our own data set as an experiment we also depict the methodology in action. But, it must be kept in mind that there are certain approximations in the final sentiment analysis, this is mainly because of the inability of present sentiment analysis modules to capture emotions like sarcasm and subtle criticism that is often adopted by the users of social media platforms.

VII. Future Scope

The quality and accuracy of social media analysis can be drastically improved if the sentiment analysis tools can capture more nuanced emotions such as sarcasm, subtle criticism, context of the conversation and the possible context of the images and links that are attached to tweets. Apart from this, the analysis in this paper can be extended by tokenizing the text of the tweets and finding out the recurring problems that the users while using each of these online food ordering services.

References

[1] Vishal A. Kharde, S.S. Sonawane *Sentiment Analysis of Twitter Data: A Survey of Techniques*, International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, April 2016

[2] Walaa Medhat, Ahmed Hassan, Hoda Korashy s*Sentiment analysis algorithms and applications: A survey*, 2014

[3] Liang-Chih Yu, Jheng-Long Wu, Pei-Chann Chang, Hsuan-Shou Chu *Sentiment analysis algorithms and applications: A surveyUsing a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news*, Knowl-Based Syst, 41 (2013)

[4] https://kleiber.me/blog/2018/02/25/top-10-python-nlp-libraries-2018/ 2018

[5] https://www.topbots.com/4-different-approaches-natural-language-processing-understanding/

[6] http://moin.ourproject.org/moin/projects/nltk/ProjectIdeas

[7] https://github.com/nltk/nltk/wiki

[8] https://elitedatascience.com/python-nlp-libraries

[9] https://www.quora.com/What-are-Natural-Language-Processing-Models
[10] Bo Pang and Lillian Lee *Opinion Mining and Sentiment Analysis* Foundations and Trends R in Information Retrieval Vol. 2, Nos. 1–2 (2008) 1–135
[11] http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/
[12] http://scikit-learn.org/stable/tutorial/basic/tutorial.html#type-casting
[13] ttps://stackoverflow.com/questions/49098726/extract-tweets-with-some-special-keywords-from-twitter-using-tweepy-in-python *search tweets on Tweepy*
[14] https://www.pythoncentral.io/how-to-use-the-twython-twitter-python-library/ *Twypython Library*
[15] https://stackoverflow.com/questions/38219889/how-to-convert-result-of-tweepy-search-to-a-json-form/38220547 *convert tweepy to JSON dumps*
[16] https://stackoverflow.com/questions/38219889/how-to-convert-result-of-tweepy-search-to-a-json-form/38220547 *An introduction to twitter-analysis*
[17] http://docs.tweepy.org/en/v3.6.0/streaming_how_tohtml *streaming via tweepy API*
[18] https://github.com/tweepy/tweepy/blob/master/tweepy/parsers.py *Tweepy Parser*
[19] https://galeascience.wordpress.com/2016/03/18/collecting-twitter-data-with-python/ *Collecting Twitter Data*
[20] https://github.com/tweepy/tweepy/issues/935
[21] https://developer.twitter.com/en/docs/tweets/search/overview/standard
[22] http://adilmoujahid.com/posts/2014/07/twitter-analytics/ *Streaming via Twitter*
[23] Wes McKinney *Python for Data Analysis*, O'Reilly, 2012
[24] https://programminghistorian.org/en/lessons/sentiment-analysis
[25] CJ. Hutto, Eric Gilbert *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text* Georgia Institute of Technology