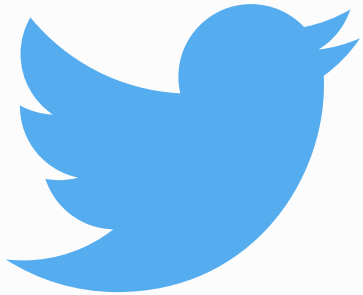
PYTHON PROGRAMMING

SENTIMENT ANALYSIS

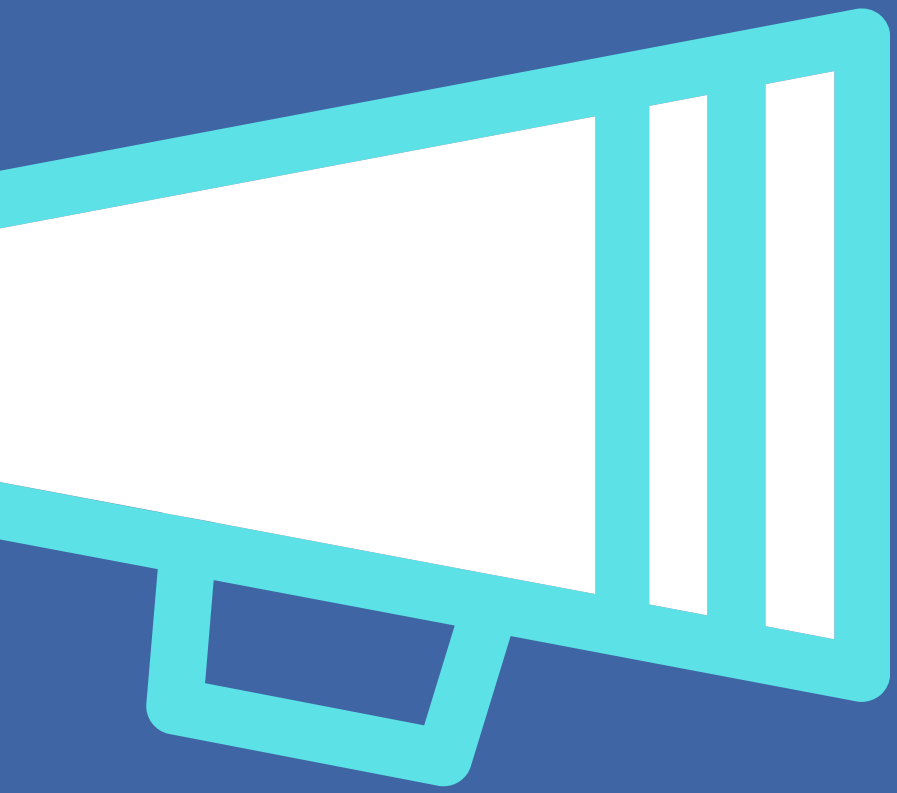


TEAM COOK

NAKAMURA KAI
XU EMELINE YISU
LI HÉLÈNE

SPRING 2019
CS159

Introduction



Sentiment analysis text mining

- Positive
- Negative
- Neutral

PREREQUISITES

TWITTER ACCOUNT → **TWITTER API** → **TWITTER LIBRARIES** → **TWITTER CREDENTIALS**

- TWITTER API: ACCESS TO DATABASE
- TWITTER LIBRARIES: TWEEPY, TEXTBLOB, NUMPY, MATPLOTLIB, PANDAS, RE
- TWITTER CREDENTIALS: KEYS & TOKENS

TWEEPY

AUTHORISE TWITTER API CLIENT

```python
class TwitterClient():
    def __init__(self, twitter_user=None):
        self.auth = TwitterAuthenticator().authenticate_twitter_app()
        self.twitter_client = API(self.auth)

        self.twitter_user = twitter_user

    def get_twitter_client_api(self):
        return self.twitter_client

    def get_user_timeline_tweets(self, num_tweets):
        tweets = []
        for tweet in Cursor(self.twitter_client.user_timeline, id=self.twitter_user).items(num_tweets):
            tweets.append(tweet)
        return tweets

    def get_friend_list(self, num_friends):
        friend_list = []
        for friend in Cursor(self.twitter_client.friends, id=self.twitter_user).items(num_friends):
            friend_list.append(friend)
        return friend_list

    def get_home_timeline_tweets(self, num_tweets):
        home_timeline_tweets = []
        for tweet in Cursor(self.twitter_client.home_timeline, id=self.twitter_user).items(num_tweets):
            home_timeline_tweets.append(tweet)
        return home_timeline_tweets
```

OAuth

1. Get a request token from Twitter
2. Redirect user to twitter.com to authorize our application
3. If using a callback, twitter will redirect the user to us. Otherwise the user must manually supply us with the verifier code.
4. Exchange the authorized request token for an access token.

```python
class TwitterAuthenticator():

    def authenticate_twitter_app(self):
        auth = OAuthHandler(twitter_credentials.CONSUMER_KEY, twitter_credentials.CONSUMER_SECRET)
        auth.set_access_token(twitter_credentials.ACCESS_TOKEN, twitter_credentials.ACCESS_TOKEN_SECRET)
        return auth
```

TWEEPY

TWITTER STREAMER

```python
class TwitterStreamer():

    def __init__(self):
        self.twitter_autenticator = TwitterAuthenticator()

    def stream_tweets(self, fetched_tweets_filename, hash_tag_list):
        # This handles Twitter authentification & connection to Twitter Streaming API
        listener = TwitterListener(fetched_tweets_filename)
        auth = self.twitter_autenticator.authenticate_twitter_app()
        stream = Stream(auth, listener)

        # This line filter Twitter Streams to capture data by the keywords:
        stream.filter(track=hash_tag_list)
```

TWITTER LISTENER

```python
class TwitterListener(StreamListener):

    def __init__(self, fetched_tweets_filename):
        self.fetched_tweets_filename = fetched_tweets_filename

    def on_data(self, data):
        try:
            print(data)
            with open(self.fetched_tweets_filename, 'a') as tf:
                tf.write(data)
            return True
        except BaseException as e:
            print("Error on_data %s" % str(e))
        return True

    def on_error(self, status):
        if status == 420:
            # Returning False on_data method in case rate limit occurs.
            return False
        print(status)
```

```python
class TweetAnalyzer():

    def clean_tweet(self, tweet):
        return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())

    def analyze_sentiment(self, tweet):
        analysis = TextBlob(self.clean_tweet(tweet))

        if analysis.sentiment.polarity > 0:
            return 1
        elif analysis.sentiment.polarity == 0:
            return 0
        else:
            return -1
```

CLEAN TEXT & ANALYSIS WITH TEXTBLOB

ORGANISE & CLASSIFY

```python
def tweets_to_data_frame(self, tweets):
    df = pd.DataFrame(data=[tweet.text for tweet in tweets], columns=['tweets'])

    df['id'] = np.array([tweet.id for tweet in tweets])
    df['len'] = np.array([len(tweet.text) for tweet in tweets])
    df['date'] = np.array([tweet.created_at for tweet in tweets])
    df['source'] = np.array([tweet.source for tweet in tweets])
    df['likes'] = np.array([tweet.favorite_count for tweet in tweets])
    df['retweets'] = np.array([tweet.retweet_count for tweet in tweets])

    return df
```

```python
if __name__ == '__main__':

    twitter_client = TwitterClient()
    tweet_analyzer = TweetAnalyzer()

    api = twitter_client.get_twitter_client_api()
    filter = "fifawwc"
    number = 200
    tweets = api.user_timeline(screen_name=filter, count=number)

    df = tweet_analyzer.tweets_to_data_frame(tweets)
    df['sentiment'] = np.array([tweet_analyzer.analyze_sentiment(tweet) for tweet in df['tweets']])
```

DATA ANALYSIS

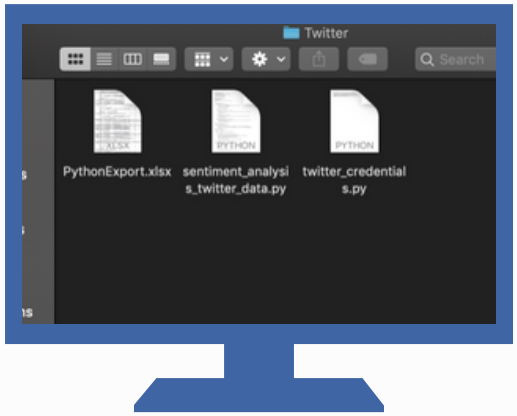
DATA VISUALISATION

EXCEL TABLE

python code

```
# Export to excel
writer = ExcelWriter('PythonExport.xlsx') # Create file
df.to_excel(writer,'Sheet1') # Transform dataframe df to excel sheet named "sheet1"
writer.save() # Save into current folder

print(df.head(10))
```



output

	tweets	id	len	date	source	likes	retweets	sentiment
0	Goooooooooood morning, football fans 🌅 Who is rea	1.13759E+18	139	2019-06-09 05:20:00	Twitter Ads Composer	266	77	1
1	First game. First goals. Highs. Lows. And emotions. #F	1.13758E+18	124	2019-06-09 04:45:00	Twitter Media Studio	91	21	1
2	#BRA's @MonicaHickmannA says there are no favouri	1.13752E+18	140	2019-06-09 00:30:00	Twitter Media Studio	311	61	0
3	✨ Giulia Gwinn ✨ @DFB_Frauen | #FIFAWWC High	1.13751E+18	118	2019-06-08 23:45:00	Twitter Media Studio	339	82	0
4	On the menu with @TheMatildas goal machine @sam	1.13748E+18	139	2019-06-08 22:01:15	Twitter Web Client	189	32	1
5	Tonight's #PlayeroftheMatch presented by @Visa for	1.13748E+18	140	2019-06-08 21:59:17	Twitter Media Studio	222	43	1
6	The first-ever goal at the #FIFAWWC for #BanyanaBan	1.13748E+18	140	2019-06-08 21:57:00	Twitter Media Studio	346	98	1
7	RT @Janinevanwyk5: Wow what a game full of emotic	1.13748E+18	139	2019-06-08 21:56:41	Twitter for iPhone	0	332	1
8	@ashlnhrris We hear ya. https://t.co/14TyiWISQg	1.13748E+18	47	2019-06-08 21:53:43	Twitter for iPhone	109	8	0
9	We got some photos did. #USA #FIFAWWC https://t.c	1.13748E+18	61	2019-06-08 21:51:42	Twitter for iPhone	1063	124	0
10	@ashlynkriegers yeah we can do that for you. https://	1.13748E+18	68	2019-06-08 21:43:40	Twitter Web Client	119	28	0
11	Alex Morgan, #USA 2019 #FIFAWWC https://t.co/Kn	1.13747E+18	81	2019-06-08 21:41:30	Twitter for iPhone	840	148	0
12	GUESS WHICH TEAM HAD THEIR #FIFAWWC SQUAD P	1.13747E+18	79	2019-06-08 21:36:11	Twitter Web Client	641	52	0
13	Ok, cool. Two secs.	1.13747E+18	19	2019-06-08 21:35:12	Twitter Web Client	318	9	1

DATA VISUALISATION

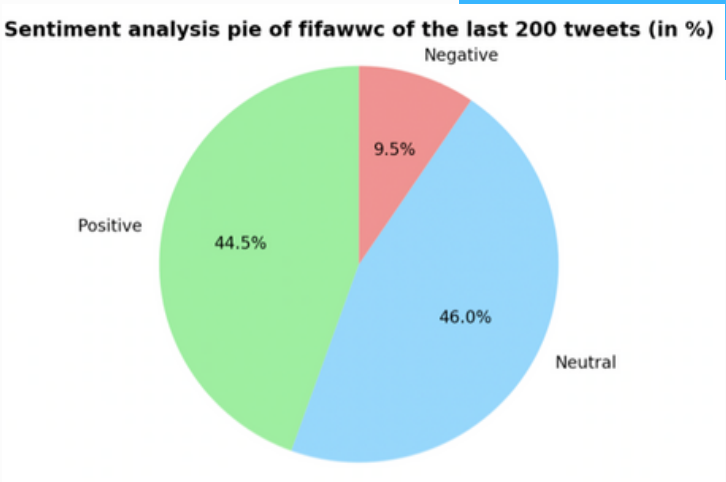
PIE CHART

python code

```
    Positive = 0
    Neutral = 0
    Negative = 0
    for index in range(len(df['sentiment'])):
        if df['sentiment'][index] == 1:
            Positive = Positive + 1
        if df['sentiment'][index] == 0:
            Neutral = Neutral + 1
        if df['sentiment'][index] == -1:
            Negative = Negative + 1

    pieLabels = 'Positive', 'Neutral', 'Negative'
    sentimentPercentage = [Positive, Neutral, Negative]
    colors = ['lightgreen', 'lightskyblue', 'lightcoral']
    figureObject, axesObject = plt.subplots()
    axesObject.pie(sentimentPercentage,labels=pieLabels,colors=colors,autopct='%1.1f%%',startangle=90)
    axesObject.axis('equal')
    plt.title("Sentiment analysis pie of " + filter + " of the last " + str(number) + " tweets (in %)",
fontweight="bold")
    plt.show()
```

output



Conclusion

EMOTIONS ARE ESSENTIAL TO EFFECTIVE COMMUNICATION BETWEEN HUMANS

thank you!

FOR YOUR TIME