

nlptweets

June 19, 2020

0.0.1 Twitter Sentiment Analysis - Gov CDMX

```
[ ]: import matplotlib.dates as mdates
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from tensorflow.python.lib.io import file_io
```

```
[ ]: !export GOOGLE_APPLICATION_CREDENTIALS="/Users/admin/Downloads/
↳gov-cdmx-twitter-sentiment-3479e766b2c0.json"
!export PATH="/usr/local/Caskroom/google-cloud-sdk/latest/google-cloud-sdk/bin:/
↳usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Library/TeX/texbin:/Users/admin/
↳development/gov-cdmx-twitter-sentiment/streaming/env/bin:/usr/local/Caskroom/
↳google-cloud-sdk/latest/google-cloud-sdk/bin:/usr/local/sbin:/Users/admin/
↳golang/bin:/usr/local/Cellar/go/1.13/libexec/bin:/Users/admin/lib:/Users/
↳admin/golang/bin:/usr/local/Cellar/go/1.13/libexec/bin:/Users/admin/lib:/
↳Library/TeX/texbin"
```

```
[ ]: !pip install --upgrade google-api-python-client
```

```
[ ]: %!bq tables describe --name gov-cdmx-twitter-sentiment:
↳gov_cdmx_twitter_sentiment.tweets
```

Now that the data is in bigquery we can save a section to cloud storage or grab it direct from bigquery.

```
[ ]: with file_io.FileIO('gs://gov-cdmx-twitter-sentiment/nlpstorage/
↳bq-results-20200618-160924-nmrdfhmw1ddj.csv', 'r') as f:
    df = pd.read_csv(f)
```

```
[ ]: df.head()
```

```
[ ]: print(df['user_text'][0])
print(len(df['user_text']))
```

```
[ ]: rts=df[df['user_text'].str.match('RT')]
print("Retweets ", len(rts['user_text']))
cdmx=df[df['user_text'].str.contains('#CDMX')]
```

```
print("#CDMX", len(cdmx['user_text']))
```

```
[ ]: from google.cloud import language
from google.cloud.language import enums
from google.cloud.language import types
from google.oauth2 import service_account
from google.protobuf.json_format import MessageToDict

score=[]
magnitude=[]
creds = service_account.Credentials.from_service_account_file('/Users/admin/
↳Downloads/gov-cdmx-twitter-sentiment-53dba8db6dbd.json')
client = language.LanguageServiceClient(credentials=creds)

for tweet in cdmx['user_text']:
    document = types.Document(
        content=tweet,
        type=enums.Document.Type.PLAIN_TEXT
    )
    analyze_sentiment_response = client.analyze_sentiment(document=document)
    message = MessageToDict(analyze_sentiment_response,
↳including_default_value_fields=True)
    score.append(message['documentSentiment']['score'])
    magnitude.append(message['documentSentiment']['magnitude'])

print(len(score))
# print('POLARITY=%s MAGNITUDE=%s for %s' % (score, magnitude, tweet))
```

```
[ ]: cdmx['score']=score
cdmx['magnitude']=magnitude
cdmx.head()
```

```
[ ]: cdmx['datef']=pd.to_datetime(cdmx['tweet_timestamp'], unit='s',
↳yearfirst='TRUE')
cdmx['datef'].head()
```

```
[ ]: #averages
score_avg=np.mean(cdmx['score'])
magnitude_avg=np.mean(cdmx['magnitude'])
print('score', score_avg, 'magnitude', magnitude_avg)
```

```
[ ]: data = pd.concat([cdmx['datef'], cdmx['magnitude']], axis=1)
data.set_index('datef',inplace=True)
fig, ax = plt.subplots(figsize=(15,7))
data.plot(ax=ax, legend=False)
ax.axhline(y=magnitude_avg, linewidth=4, color='r')
ax.xaxis.set_major_locator(mdates.MinuteLocator(interval=5))
```

```
ax.xaxis.set_major_formatter(mdates.DateFormatter('%M'))
ax.set_xlabel('Date July 10th ~3:30 P.M')
ax.set_ylabel('Magnitude')
```

```
[ ]: data2 = pd.concat([cdmx['datef'], cdmx['score']], axis=1)
data2.set_index('datef', inplace=True)
fig, ax2 = plt.subplots(figsize=(15,7))
data2.plot(ax=ax2, legend=False)
ax2.axhline(y=score_avg, linewidth=4, color='r')
ax2.xaxis.set_major_locator(mdates.MinuteLocator(interval=5))
ax2.xaxis.set_major_formatter(mdates.DateFormatter('%M'))
ax2.set_xlabel('Date July 10th ~3:30 P.M')
ax2.set_ylabel('Score')
```

High magnitude tweets are more impactful than low magnitude tweets, a weak statement doesn't say much. We will concentrate only on statements with a magnitude of 0.5 or higher.

```
[ ]: fig, ax = plt.subplots()
ax.figure.set_size_inches(10,4)
ax.grid(False)
ax.scatter(cdmx.magnitude, cdmx.score, s=120, c='black', alpha=0.5)
ax.set(xlabel='magnitude', ylabel='score')
plt.show()
```

We can also see that things around zero polarity (neither very positive nor negative) are not interesting to flag

```
[ ]: love=cdmx[(cdmx['magnitude'] >=0.5) & (cdmx['score'] >=0.5)]
hate=cdmx[(cdmx['magnitude'] >=0.5) & (cdmx['score'] <= -0.5)]
```

```
[ ]: print(len(cdmx['magnitude']), len(love['magnitude']), len(hate['magnitude']))
print("weak", len(cdmx['magnitude']) -
      → (len(love['magnitude'])+len(hate['magnitude'])))
```

```
[ ]: labels = ['loves', 'hates', 'weak']
sizes = [7, 0, 38]
colors = ['gold', 'lightskyblue', 'lightcoral']
patches, texts = plt.pie(sizes, colors=colors, shadow=True, startangle=90)
plt.legend(patches, labels, loc="best")
plt.axis('equal')
plt.tight_layout()
plt.show()
```

```
[ ]: fig, ax= plt.subplots()
ax.scatter(love.magnitude, love.score, s=120, c='purple', alpha=0.5)
ax.scatter(hate.magnitude, hate.score, s=120, c='red', alpha=0.5)
ax.figure.set_size_inches(10,4)
```

```
ax.grid(False)
ax.set_xlabel('magnitude >0.5', ylabel='abs(score) >=0.5')
plt.show()
```

```
[ ]: fig, ax1 = plt.subplots()
ax1.set_xlabel('Date July 10th ~3:30 P.M')
ax1.set_ylabel('magnitude')
ax1.grid(False)
ax1.xaxis.set_major_locator(mdates.MinuteLocator(interval=5))
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%M'))
ax1.figure.set_size_inches(10,4)
ax1.plot(love['datef'], love['magnitude'], 'g*', markersize=20, alpha=0.5)
ax1.plot(hate['datef'], hate['magnitude'], 'r*', markersize=20, alpha=0.5)
ax2=ax1.twinx()
ax2.grid(False)
ax2.plot(love.datef, love.score, 'g.', markersize=20, alpha=0.5)
ax2.plot(hate.datef, hate.score, 'r.', markersize=20, alpha=0.5)
ax2.set_ylabel('score')
plt.show()
```

```
[ ]: fig, ax = plt.subplots()
ax.set_xlabel('Date July 10th ~3:30 P.M')
ax.grid(False)
ax.xaxis.set_major_locator(mdates.MinuteLocator(interval=5))
ax.xaxis.set_major_formatter(mdates.DateFormatter('%M'))
ax.figure.set_size_inches(10,4)
ax.plot(love['datef'], love['magnitude'], 'g', linestyle=':', marker='*',
    ↳linewidth=2, markersize=20, alpha=0.5)
ax.plot(hate['datef'], hate['magnitude'], 'k', linestyle=':', marker='.',
    ↳linewidth=2, markersize=20, alpha=0.5)
ax.set_ylabel('magnitude')
plt.show()
```

```
[ ]: fig, ax = plt.subplots()
ax.set_xlabel('Date July 10th ~3:30 P.M')
ax.set_ylabel('score')
ax.grid(False)
ax.xaxis.set_major_locator(mdates.MinuteLocator(interval=5))
ax.xaxis.set_major_formatter(mdates.DateFormatter('%M'))
ax.figure.set_size_inches(10,4)
ax.plot(love['datef'], love['score'], 'b', linestyle=':', marker='.',
    ↳linewidth=2, markersize=20, alpha=0.5)
ax.plot(hate['datef'], hate['score'], 'r', linestyle=':', marker='.',
    ↳linewidth=2, markersize=20, alpha=0.5)
plt.show()
```

```
[ ]:
```