SD-Pràctica 2

Recollida i tractament de dades

Video: https://youtu.be/_45G2h8ysFA

GitHub: https://github.com/Zackon123-afk/SD-Pract2.git

Curs 2020/2021

Alex Soriano Faiges

Arnau Francesc Llaberia Declara

Índex

1. Recollida de informació

Per a recollir la informació de internet, hem utilitzat Twitter, amb la utilització de la llibreria Tweepy.

1.1. Tweepy

Primer de tot, per a utilitzar tweepy, que no deixa de ser la API que ens dona twitter per a la recollida d'informació d'aquest, hem de donar d'alta com developers el nostre compte; així aconseguirem les nostres claus d'accés que després utilitza per a fer servir la llibreria.

```
def get_auth():
    auth = tweepy.OAuthHandler("", "")
    auth.set_access_token("", "")
    return auth
```

Després d'autenticar-nos, utilitzem la funció *tweepy.search* per buscar els tweets amb la paraula passada per paràmetre i el idioma que volem. (inicialitzem el *SentimentIntensityAnalyzer*).

```
def tweepy_scan(word):
    global BUCKET

    auth = get_auth()
    api = tweepy.API(auth, wait_on_rate_limit=True)

    analyzer = SentimentIntensityAnalyzer()
```

Per cada vacuna amb els seus tweets, guardem en un diccionari: el text, la url, el sentiment del tweet (mitjançant la llibreria vaderSentiment, on primer fem la traducció al anglés amb la llibreria *mtranslate*), la data i el lloc d'on es fa. ho comprimim en format json amb el nom de la paraula que es ha buscat, ho pengem al bucket de IBM.

```
datos = {
    "Mensaje": [],
    "url": [],
    "sentiment": [],
    "date": [],
    "local":[],
}

for status in tweepy.Cursor(api.search, q=word, lang="es", tweet_mode="extended").items(250): #numberOftwets
    datos["Mensaje"].append(status.full_text)
    datos["url"].append("https://twitter.com/twitter/statuses/"+str(status.id)+",")
    datos["date"].append(status.created_at.strftime("%m/%d/%Y %H:%M:%S"))
    datos["local"].append(str(status.user.location))


for text in datos["Mensaje"]:
    string_twi = mtranslate.translate(str(text),"en", "auto")
    datos["sentiment"].append(str(analyzer.polarity_scores(string_twi)['compound']))

now = datetime.now()
data = now.strftime("%m/%d/%Y")

storage = Storage()
storage.put_object(BUCKET,data+"-"+word+".json",json.dumps(datos))
```

2. Tractament de dades

Per cada fitxer .json el llegim i generem un diccionari de dades per tal de després poder fer les gràfiques amb aquestes.

```
def datos_twitter(word):
    global BUCKET

    datos_grafi = {
        "sent_pos" : 0,
        "sent_neg" : 0,
        "mitjana": 0.0,
        "sent_hist": [],
        "word": word,
        "location_count": {
        }
    }

    now = datetime.now()
    data = now.strftime("%m/%d/%Y")
    key= data+"-"+ word +'.json'

    storage = Storage()
    json_read = storage.get_object(BUCKET,key)
    data = json.loads(json_read)

    mitjana = 0
    for sent in data["sentiment"]:
        sent = float(sent)
        if sent >= 0:
            datos_grafi["sent_pos"] += 1
        else:
            datos_grafi["sent_neg"] += 1

        mitjana += sent
        datos_grafi["sent_hist"].append(round(sent, 1))

        datos_grafi["mitjana"] = (mitjana/len(data["sentiment"]))

    for loc in data["local"]:
        if loc in datos_grafi["location_count"]:
            datos_grafi["location_count"][loc] += 1
        else:
            datos_grafi["location_count"][loc] = 1

    return datos_grafi
```

Com es pot observar, guardem el numero de tweets amb sentiment positiu, sentiment negatiu, la mitjana, tots els sentiments en una array, la paraula que te aquest diccionari i finalment tenim un altre diccionari que guarda les localitzacions i el nombre de cops que si ha fet un tweet desde aquestes.

Calculem 4 gràfics, utilitzan la llibreria *matplotlib*:

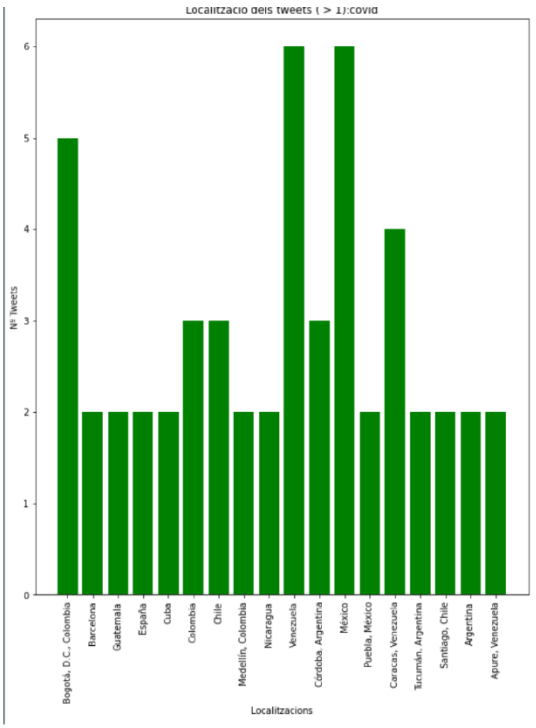
El primer (**gràfic_localitzacio**) mostra els llocs on s'han fet 2 tweets o més en un graf de barres, per tal de poder comparar desde quina localització es parla més sobre aquella paraula.

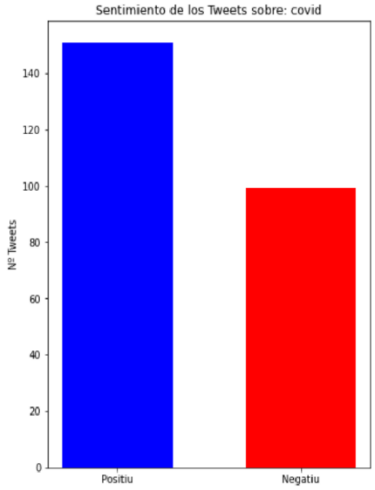
El segon (**gràfic_sentiment**) comparem la quantitat de sentiments positius en contra dels negatius per paraula.

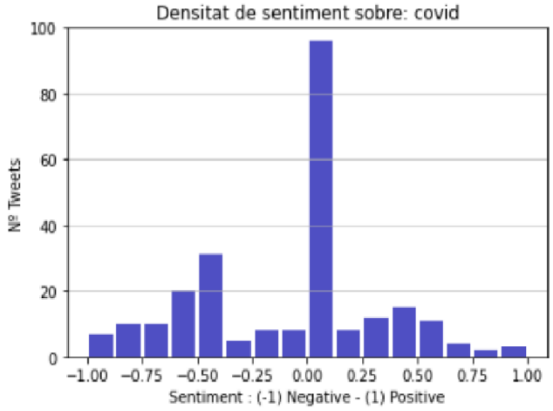
El tercer (**gràfic_historiaSentiment**) es un histograma sobre el sentiments dels tweets, representan així el numero de vegades que ocurreix una determinada puntuació de sentiment.

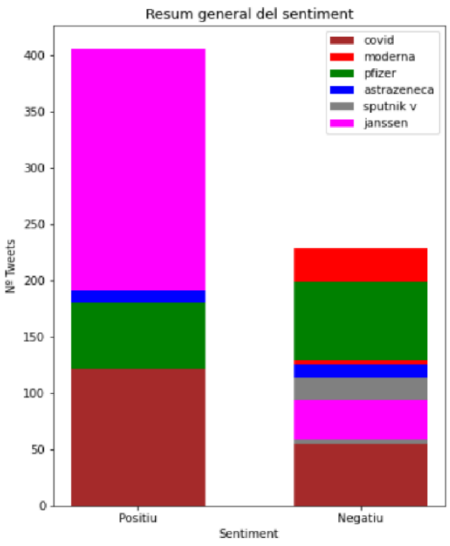
El quart (**gràfic_globalSentiment**), fa el mateix que el **gràfic_sentiment** pero ho fa amb totes les paraules juntes.

Exemples:









3. Execució al núvol

3.1. Tweepy

Per a executar al núvol, agafem cada un de les funcions principals i amb el *pool.map* executem al núvol. Primer realitzem el **tweepy_scan** per tal de generar un json per paraula al ibm cloud i despres amb **datos_twitter** per generar els diferents diccionaris agafant els fitxers json del cloud.

Cada tweepy_scan recull la informació de 250 tweets per tant en total de totes les paraules es de 1500 tweets.

Despues utilitzant els diccionaris generem les gràfiques.

```
if __name__ == '__main__':


    with Pool() as pool:
        inicio = time.time()
        pool.map( tweepy_scan, [ "covid", "moderna", "pfizer", "astrazeneca", "sputnik v", "janssen"])
        fin = time.time()
        inicio2 = time.time()
        datos = pool.map( datos_twitter, [ "covid", "moderna", "pfizer", "astrazeneca","sputnik v", "janssen"])
        fin2 = time.time()

    print("Temps en segons:")
    print("Tiempo de tweepy_scan: " + str(fin-inicio))
    print("Tiempo de datos twitter:" + str(fin2-inicio2))
    print("Tiempo Total: " + str((fin-inicio) + (fin2-inicio2)))

    grafic_localitzacio(datos[0])
    grafic_localitzacio(datos[1])
    grafic_localitzacio(datos[2])
    grafic_localitzacio(datos[3])
    grafic_localitzacio(datos[4])
    grafic_localitzacio(datos[5])

    grafic_sentiment(datos[0])
    grafic_sentiment(datos[1])
    grafic_sentiment(datos[2])
    grafic_sentiment(datos[3])
    grafic_sentiment(datos[4])
    grafic_sentiment(datos[5])

    grafic_historiaSentiment(datos[0])
    grafic_historiaSentiment(datos[1])
    grafic_historiaSentiment(datos[2])
    grafic_historiaSentiment(datos[3])
    grafic_historiaSentiment(datos[4])
    grafic_historiaSentiment(datos[5])

    grafic_globalSentiment(datos)
```

Aquí podem vore el que sol tardar, tot i que depèn de si hem fet moltes quèries amb el tweepy anteriorment i el estat del cloud.

```
Temps en segons:
Tiempo de tweepy_scan: 88.42315196990967
Tiempo de datos twitter:3.4190986156463623
Tiempo Total: 91.84225058555603
```