Note

Riccardo Ferrarese

19/3/2021

Text mining and Network Analysis of

Setup App

- installare *git*, sistema per il controllo del versionamento
- installare *renv*, permette di creare un ambiente di esecuzione isolato per la gestione delle dipendenze. Permette anche di utilizzare il pacchetto *reticulate* per integrare degli script in Python e gestire l'ambiente locale per la loro esecuzione.
- creare un progetto *shiny app*

Reddit

Reddit, a differenza della maggior parte dei social-network maggiormente utilizzati, non presenta un meccanismo per scegliere una cerchia di utenti con cui interagire (per intenderci, i followers su Twitter o Instagram), tuttavia il modo in cui è organizzato permette all'utente di interagire in diversi forums, ognuno di uno specifico argomento, nel quale interagirà con gli altri utenti che 'seguo' lo stesso *subreddit*. Inoltre sulla pagina generale si possono trovare i post più popolari di tutta la piattaforma.

Si può definire Reddit come un *social-network aggregator*, cioè è una piattaforma di discussione in cui gli utenti possono discutere o condividere informazioni, suddivisa in forums di specifici argomenti chiamati **subreddits**.

Ogni utente può eseguire un numero illimitato di interazioni con la piattaforma, dove le interazioni possono essere la scrittura di un post in uno specifico subreddit, commentare post e interazioni di altri utenti o esprimere la propria preferenza relativa a un certo post o commento. Se un certo post (a meno che non sia di un subreddit privato) riceve molti *downvotes* immediatamente crollerà sulla 'classifica dei post' e scoparirà dalla vista degli altri utenti. Al contrario se acquisisce una certa importanza potrà esser visualizzato nella pagina generale di reddit e raggiungendo così un numero maggiore di utenti.

Un ulteriore particolarità di questa piattaforma è che gli utenti sono allo stesso modo creatore di contenuti, consumatori e curatori delle informazioni in esso. Utilizza infatti un sistema a punteggio, tramite *upvotes* e *downvotes* da parte degli utenti, per determinare i contenuti e le discussioni con maggior interesse e che verrano mostrati nei primi contenuti della pagina.

Pur permettendo agli utenti di mantenere l'anonimicità utilizzando un nome utente a piacere, Reddit tiene traccia di tutte le attività di ogni profilo inclusi i post e i commenti effettuati.

Collect Data with Python

Per la raccolta dati è stato utilizzato Python dal momento che è presente la libreria *Pushshift* per l'estrazione dei dati di Reddit senza vincoli stringenti sulle richieste effettuate. E' stato utilizzato un wrapper delle Python

Reddit API, **pmaw**, che permette di eseguire lo scrapping dei dati utilizzando il multithreading, in modo tale da rendere più efficente il processo di raccolta dei dati.

Le API mettono a disposizione una serie di funzioni per la ricerca dei post d'interesse, potendo specificare l'intervallo temporale e i subreddit in cui eseguire la ricerca. Inoltre, volendo estrarre da Reddit le informazioni relative a diversi subreddit è stata utilizzata la libreria *multiprocessing* per lanciare un *pool* di processi incaricati di eseguire la stessa funzione su dati diversi.

```
LIST_of_SUBREDDIT = [ 'dogecoin',
    'pancakeswap',
    'eth',
    'ethereum',
    'elon',
    'wallstreetbets']

    start_epoch=int(dt.datetime(2020, 1, 1).timestamp())
    end_epoch=int(dt.datetime(2021, 2, 1).timestamp())

    nargs = []
    for x in LIST_of_SUBREDDIT:
        nargs.append( [start_epoch, end_epoch, 'submissions', x] )
        nargs.append( [start_epoch, end_epoch, 'comments', x]  )

    pool = Pool(16)
    pool.map(run, nargs)
    pool.close()
    pool.join()
```

La funzione *run* lanciata da *pool.map* istanzia un oggetto di classe Miner contente la chiave di autenticazione per accedere a Pushshift e i metadati delle informazioni da raccogliere. In questo modo vengono lanciati diversi **Miner**, ognuno per uno specifico subreddit, i quali eseguono la ricerca e il salvataggio dei dati in parallelo.

```
def run(args) -> pd.DataFrame:
    """
    Function for starts miner's' process

    Args:
        [start, end]: [temporal intervall where we would scrap data]
        [item]: element to scrap
    Returns:
        [type]: [description]
    """

    print(args)
    start, end, item, subreddit = args

    miner = Miner(start, end, item, subreddit)
    miner.perform_search()
    return miner.read_data()

class Miner(object):
    """ Class for Reddit Data Mining"""
```

```
    def __init__(self, start_epoch, end_epoch, func, subreddit) -> None:
        super().__init__()
        self.api = PushshiftAPI(rate_limit=100)
        self.start_time = start_epoch
        self.end_time= end_epoch
        self.subreddit = subreddit
        self.data = None
        self.func = func

    def read_data(self):
        return self.data

    def perform_search(self):
        item = self.func
        print(f'Start search {item}...')
        if item == 'submissions':
            df = self.search_save_sub(self.subreddit)
            self.data = df
        if item == 'comments':
            df = self.search_save_com(self.subreddit)
            self.data = df

    @timeit
    def search_save_sub(self, subreddit):
        api = self.api
        res_ = api.search_submissions(after=self.start_time,
                                  before=self.end_time,
                                  subreddit=subreddit,
                                  filter=COLS_SUB,
                                  #limit=2
                                  )
        data = pd.DataFrame([x for x in res_])
        data.to_csv(f"./data/{self.subreddit}_sub.csv")
        print(f"write {self.subreddit}_sub.csv")

    @timeit
    def search_save_com(self, subreddit):
        api = self.api
        res_ = api = self.api.search_comments(after=self.start_time,
                                  before=self.end_time,
                                  subreddit=subreddit,
                                  filter=COLS_COM,
                                  #limit=2
                              )
        data = pd.DataFrame([x for x in res_])
        data.to_csv(f"./data/{self.subreddit}_com.csv")
        print(f"write {self.subreddit}_com.csv")
```

E' stato utilizzato un *rate-limit* per le richieste all'API pari a 100, leggermente superiore al limite di default ma leggermente inferiore al limite imposto dalle richieste al minuto massime che si possono effettuare per la raccolta.

La funzione *decorator timeit* permette inoltre di avere una misura indicativa del tempo impiegato per ogni

richiesta. In totale l'esecuzione ha impiegato . . . , ma essendo eseguita in parallelo il tempo reale di esecuzione è stato di un paio di ore.

Di seguito è mostrato un estratto di output da terminale che si ottiene dopo l'esecuzione del programma.

```
args first process: [1577833200, 1614553200, 'submissions']
args second process: [1577833200, 1614553200, 'comments']
Start search comments...
Start search submissions...
1430174 results available in Pushshift
259899 results available in Pushshift
Checkpoint:: Success Rate: 36.00% - Requests: 100 - Batches: 10 - Items Remaining: 256299
Checkpoint:: Success Rate: 29.00% - Requests: 100 - Batches: 10 - Items Remaining: 1427274
Checkpoint:: Success Rate: 35.50% - Requests: 200 - Batches: 20 - Items Remaining: 253225
Checkpoint:: Success Rate: 27.50% - Requests: 200 - Batches: 20 - Items Remaining: 1424674
Checkpoint:: Success Rate: 33.00% - Requests: 300 - Batches: 30 - Items Remaining: 250824
Checkpoint:: Success Rate: 29.33% - Requests: 300 - Batches: 30 - Items Remaining: 1421374
Checkpoint:: Success Rate: 31.00% - Requests: 400 - Batches: 40 - Items Remaining: 248435
Checkpoint:: Success Rate: 30.75% - Requests: 400 - Batches: 40 - Items Remaining: 1417879

[...]

write ./doge_sub.csv
time: 194798997.85995483 ~~ 3,25h
```

Shiny App

A reactive expression is an R expression that uses widget input and returns a value. The reactive expression will update this value whenever the original widget changes.

Reddit è una piattaforma social nella quale gli utenti possono scrivere un post e gli altri possono esprimere il loro gradimento o meno, ed eventualmente possono anche interagire commentando. Reddit è un aggregatore di contenuti per specifiche comunità,

subreddit:

- DogeCoin: - doge - dogecoin
- Eth - EhtTrader -
- WallStreetBets - qua dobbiamo filtrare i commenti con le parole chiave per ciascuno dominio che ci interessa - avremmo più tab per ciascuno dei temi
- OpenSea ?? r/opensea

Data Mining

Overview of dataset

Set up data

Clean data

Prepare Data

Data Analysis

Network Data

Network Analysis

Text Mining

Results

Conclusion