Guida operativa

Federico Luzzi, Marco Peracchi, Christian Uccheddu, Gabriele Centemeri (TTC)

Di seguito la guida operativa per eseguire il codice e replicare i risultati ottenuti:

Presa dati Dicembre-Gennaio

Eseguire il file che esegue richieste ogni 30 minuti e salva i dati in formato csv.

`scraper/scraper_csv.py`

Il jupyter-notebook serve a selezionare prese dati ogni 6 ore, e si trova all'interno della cartella *clean_store_data*.

`clean_jen_video/fix_jen_json.ipynb`

Per trasformare i dati da formato csv a json.

`csv_to_json/main.py`

Lo script ha bisogno del parametro:

- -d "directory_dei_dati"

In tal modo si indica la directory dove sono i dati da trasformare. Per caricare i json su MongoDB avviare lo script:

`json_to_mongo(windows)/json_to_mongo.py`

A questo script vanno forniti i seguenti parametri:

- -d "directory dei dati"
- -u "utente mongo"
- -p "password utente"
- -port "porta in cui è attivo l'utente"
- -db "Nome del database in output"
- -c "collection in cui vengono inseriti i dati"

Presa dati Marzo-Maggio

Aprire il servizio Mongo da terminale e successivamente avviare in due terminali separati gli script:

- `scraper/scraper_consumer.py`
- `scraper/scraper_producer.py`

Mediante l'utilizzo del servizio Kafka vengono rilevati i dati ogni 6 ore.

Presa dati Covid

Scaricare i dati in formato csv da OurWorldInData ed eseguire il codice:

`covid/cleaner.py`

Questo script permette di eseguire una pulizia dei dati in modo da renderli integrabili con i json dei video precedenti.

Integrazione dei dati

Per eseguire l'integrazione tra i dati Covid e i dati di Youtube bisogna eseguire il seguente script:

`json_to_mongo(windows)/merge_to_mongo.py`

A questo script vanno forniti i seguenti parametri:

- -d "directory dei dati"
- -u "utente mongo"
- -p "password utente"
- -port "porta in cui è attivo l'utente"
- -db "Nome del database in output"
- -c "collection in cui vengono inseriti i dati"

I dati vengono quindi integrati sulla data e il paese e successivamente caricati su Mongo.

Query mongo

La seconda domanda di ricerca ci chiede di distinguere quali video riguardino il coronavirus o meno. L'espressione regolare seguente analizza tags e titoli per verificare se è presente una parola riferita alla pandemia.

```
/(corona|covid|virus|pandemi[aec]|epidemi[aec]|tampon[ei]*|sierologico|mascherin[ae]|코로나 바
이러스|fase\s*(2|due)|iorestoacasa|stayathome|lockdown|[qc]uar[ae]nt[äae]i*n
[ea]|कोरोनावाइरस|वैश्विक महामारी|massisolation|distanziamento\s*sociale|social\s*distancing|감염병 세계적 유행
|パンデミック|コロナウイルス|सर्वव्यापी महामारी|सतर्कता भारी|пандемия|коронавирус|social\s*distancing|
distanciamiento\s*social|코로나|कोविड|वैश्विक|vaccin[oe]*|isolamento|intensiv[ao]|assembrament[io]|
guant[oi]|dpi|disinfettante|swabs|emergenza|emergency|droplets*|aerosol|isolation|intensive
\s*care|crowd|gloves*|disinfectant|감염병 유행|완충기|마스크|나는 집에있어|폐쇄|사회적 거리두기|백신|모임|비상 사
태|비말|봄 혈증|écouvillon|masques*|restealamaison|confin[ae]mento*|distanciation\s*sociale|
soins\s*intensifs|rassemblements|désinfectant|urgence|gouttelettes|飛沫|タンポン|マスケリン|封鎖|
人混みを避ける|ワクチン|隔離|集会|集中治療|緊急|बूंदें|फाहे|मास्क|लॉकडाउन|सोशल डिस्टन्सिंग|टीका|गहन देखभाल|समारोहों|आपातकालीन|
gotas|cotonetes|m[áa]scaras|ficoemcasa|vac[iu]na|reuni[õo]n*es|emerg[êe]ncia|капли|тампоны|
маски|карантин|социальное\s*дистанцирование|вакцина|интенсивная\s*терапия|сходы|
чрезвычайное\s*происшествие|hisopos|mequedoencasa|cierre|Tröpfchen|Tupfer|Masken|
bleibezuHause|Ausgangssperre|soziale\s*Distanzierung|Impfstoff|Intensivstation|
Versammlungen|Notfall|건강\s*격리|検疫|संगरोध|[кК]арантин|hand|man[io]s*|소유|手|[Pp]уки|Hände|
mãos)/i
```

Figura 1: Regular expression usata

Eseguire ora i seguenti comandi all'interno della shell di MongoDB.
Creare due nuovi campi chiamati **covid_title** e **covid_tags** inizializzati entrambi a **False**

```
db.video_merge.update({},{$set : {covid_tags : false,
                        covid_title : false}},
                        {multi : true})
```

Eseguire le seguenti due query che controllano se l'espressione regolare è presente nel campo title o in uno dei tag per ogni video.

```
db.video_merge.update({tags : {$in : [REGEX]}},
                        {$set : {covid_tags: true}},
                        {multi : true})
```

```
db.video_merge.update({title : {$in : [REGEX]}},
                        {$set : {covid_title: true}},
                        {multi : true})
```

Sharding

Lo sharding dei documenti di Mongo è stato eseguito in locale, quindi utilizzando *localhost* come host. All'interno della cartella *sharding* è possibile visualizzare le cartelle contenenti i vari file di configurazione per tutti i componenti.

- **configsvr** Sono tre istanze di *mongod*, configurate come replica set. Il *config server* conosce dove ogni dato è allocato dei vari shard, quindi è importante configurarlo come replica-set, così che in caso di guasti non si perdano le informazioni.
- **router** È un'istanza di *mongos*. Per interrogare i vari shard è necessario interfacciarsi con essa.
- **shard** Sono tre istanze di *mongod*. Ogni shard è configurato in replica-set, e i dati vengono suddivisi nei vari shard.

All'interno di ogni cartella sono presenti i vari file di configurazione, dove deve essere specificato il percorso della cartella *data* di ogni istanza. Per prima cosa inizializzare i replica-set, e successivamente avviarli come shard server o come config server. Una volta collegati quest'ultimi al router è necessario caricare i dati su uno degli shard. Prima di effettuare la procedura va specificata la *shard key*, e va anche specificato il metodo di suddivisione dei dati, se *hashed* o *range*. In questo lavoro è stato utilizzato il primo metodo.