Tweets Simple Analysis Platform

Christothea Stylianidou

Table of Contents

Introduction

Implementation of a simple 3-tier web application which provides Twitter-->Tweets collection, for specified list of keywords and/or user names (aka screen_name) and basic data statistics.

Note that the current implementation is not a high-scale production ready solution. It can be used only for Proof of Concept purposes.

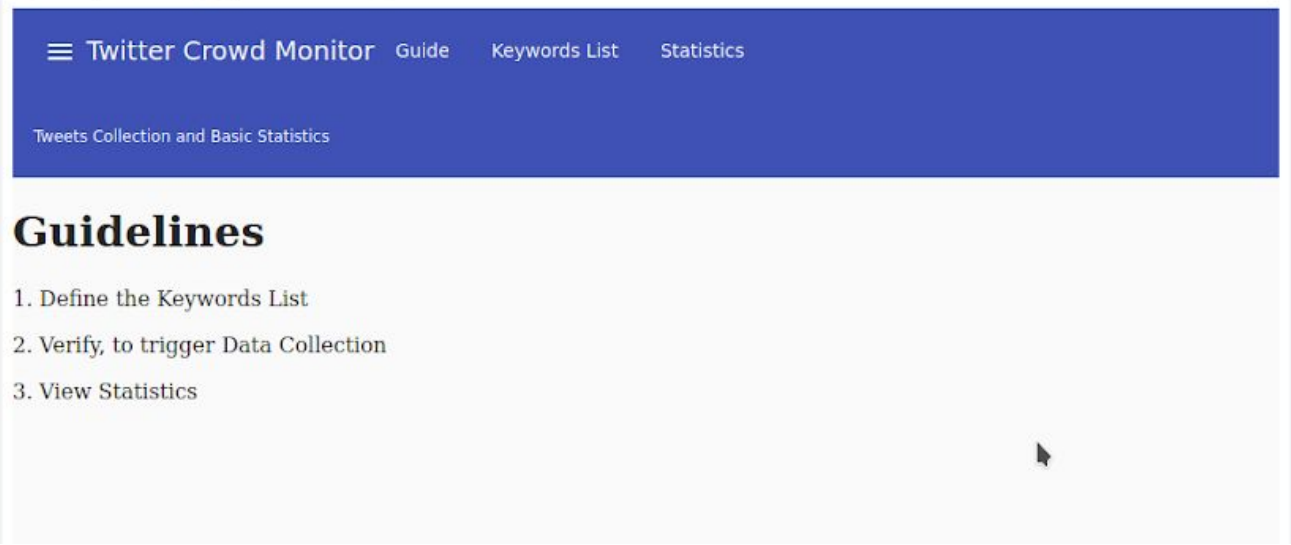
Application Details

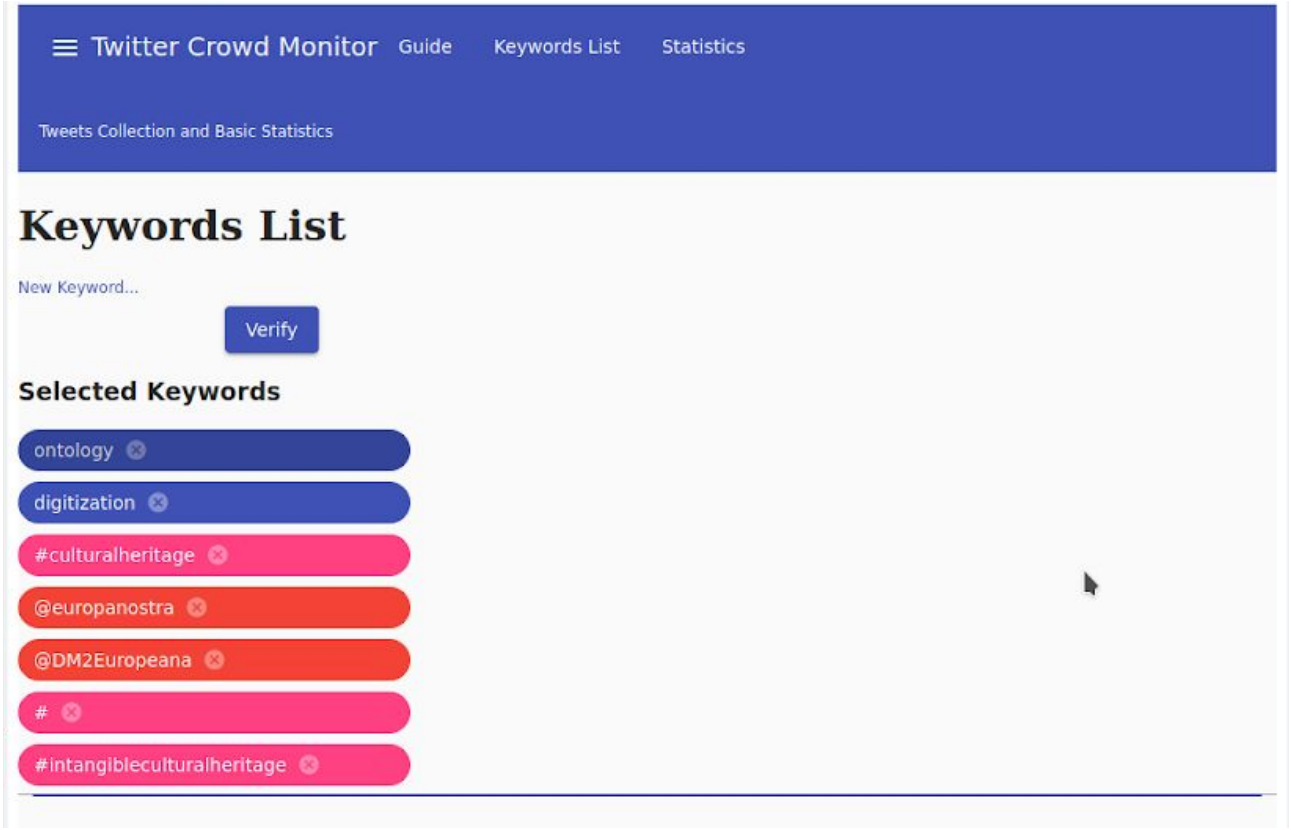
Client

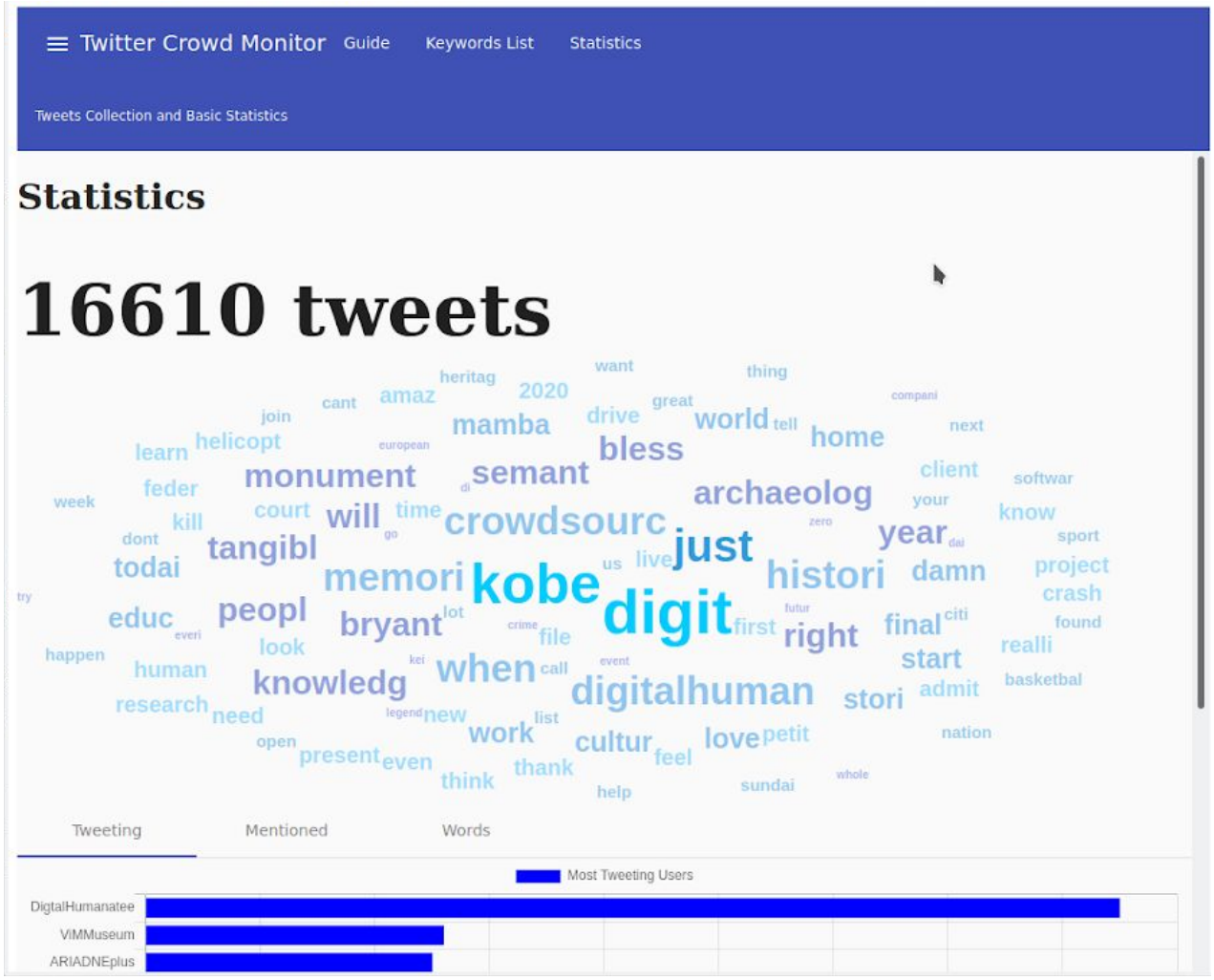
Provides basic functionality for entering a list of keywords, verify and request to view statistics:

1. Navigate to **Keywords**
    a. Type a word and press enter to be added in the list
    b. To search for the tweets for a user, prefix the user name with the character '@'
2. Once you have the list of keywords ready, Click on **Verify** to trigger collection of tweets on the backend server
3. Wait for some minutes

4. Go on Statistics to view the results:
    a. **Tweeting**: displays the top 20 users who are tweeting the most
    b. **Mentioned**: displays the top 20 users who are mentioned the most
    c. **Words**: displays the top 20 used words. In addition to the chart, you can refer to the words cloud




Twitter Crowd Monitor
Guide
Keywords List
Statistics
Tweets Collection and Basic Statistics
Keywords List
New Keyword...
Verify
Selected Keywords
ontology
digitization
#culturalheritage
@europanostra
@DM2Europeana
#
#intangibleculturalheritage


Twitter Crowd Monitor
Guide
Keywords List
Statistics
Tweets Collection and Basic Statistics
Statistics
16610 tweets
heritag
want
thing
2020
amaz
cant
great
compani
join
drive
world
tell
home
next
mamba
helicopt
learn
european
bless
monument
semant
client
softwar
feder
archaeolog
your
week
court
will
time
crowdsourc
zero
year
know
kill
just
dont
go
tangibl
live
sport
todai
us
histori
damn
project
memori
kobe
crash
try
peopl
digit
futur
educ
bryant
lot
crime
first
right
final
citi
found
everi
file
look
realli
happen
when
call
event
start
human
knowledg
digitalhuman
admit
basketbal
research
need
legend
new
list
stori
work
cultur
love
petit
nation
open
present
even
feel
think
thank
whole
help
sundai
Tweeting
Mentioned
Words
Most Tweeting Users
DigtalHumanatee
ViMMuseum
ARIADNEplus

Server

- Exposes two REST endpoints for serving clients requests:
    a. **Collect** (Post request on 'http://hostname:8009/tw-manager/collect'). This request will just trigger the collection of tweets
    b. **Statistics** (Get request on 'http://hostname:8009/tw-manager/statistics'). This request will return to the client the current state of the Summary Statistics (Tweeting, Mentioned, WordsCount, TotalTweetsNumber)
- Tweets collection Workflow:
    a. Application Controller receives **Collect Request**
    b. Trigger **Collect Tweets Event**
    c. **Tweets Collector** receives the Event
        i. Fetch data from twitter
        ii. Trigger **Collected Tweet Event**
    d. **Data analyzer** receives the Event
        i. Calculates Tweet Statistics (word, hashtags, mentions count)
            - Clean tweet text from urls
            - Tokenize
            - Remove stop words
            - Apply simple stemming
        ii. Calculates Summary Statistics
        iii. Trigger **Store Data Events** (TweetData, SummaryStatistics)

e. **Cache Service** receives the Event and stores data in data store

Tweets Collector Sample Code

```
// Conenct
this.apiClient = new twit({
                consumer_key: this.envConfig.TWAPI_CONSUMER_KEY,
                consumer_secret: this.envConfig.TWAPI_CONSUMER_SECRET,
                access_token: this.envConfig.TWAPI_ACCESS_TOKEN,
                access_token_secret: this.envConfig.TWAPI_ACCESS_TOKEN_SECRET,
                timeout_ms: this.envConfig.TWAPI_TIMEOUT_MS,
                strictSSL: this.envConfig.TWAPI_STRICT_SSL,
            });

// Fetch Data
// Endpoints:
// TWITTER_TWEETS = 'statuses/user_timeline',
// TWITTER_TWEETS_SEARCH = 'search/tweets',

const response = await this.apiClient.get(endpointType, params).catch((e) => {
                console.error('Error on Twitter API Fetch Request');
                console.error(params);
                console.error(e.stack);
            });
```

```
            if (!response || !response.resp) {
                console.error(`Error Twitter API Fetch Request Failed: General Error`);
                console.error(params);
            } else if (response.resp.statusCode === 200) {
                return response.data;
            } else if (response.resp.statusCode === 429) {                console.warn(`Warning Twitter API
Reached Requests Limit: ${response.resp.statusCode}`);
                console.warn(params);
            } else {
                console.error(`Error Twitter API Fetch Request Failed: ${response.resp.statusCode}`);
                console.error(params);
            }
```

Data Store

Data Store is accessed only from server-side and is used for basic storage:

- Tweets Data
- Summary Statistics

Setup Requirements

- **MongoDB**, at least v3.6.8
- **Node**, at least v10.18.1

- **Npm**, at least v6.13.4
- **Angular**, at least v8.3.23
- **Twitter DEV access tokens**

Notes:

- Implementation and testing had been done only on **Linux** Kubuntu v19.04. Client side tested only on firefox browser.
- For connection to the Twitter API, used ready npm package (twit) which wraps the native calls to the Twitter API.

Future Improvements

- **On the simple POC implementation**
  - Add Authentication/ Authorization and secured communication
  - Store users collect requests and keywords
  - Currently we give statistics only on all the collected tweets. Introduce Profiling to allow statistics on specific filters (keywords, usernames, ...)
  - Allow for more dynamic filtering (AND, OR operators)
  - Add connection to the Twitter Live Stream endpoint… provide real time monitoring and statistics
  - Send notifications from server to client about the state of the data collection
  - Modify the server in case of Twitter API failures (e.g. Rate limiting) to keep state and repeat the failed request
  - Add connections to other twitter endpoints as well (e.g. followers, followings, ...)
  - Generalize to be used for other data collections as well (e.g. instagram, facebook, scrapping of other sites)
  - Generalize and define our own Data Models, transform twitter and/or other collectors data into our own
  - Create links between data, basic links analysis and visualization on the client side

    - Improve client side appearance
    - On server side use node clustering ability to distribute processing to more processor cores
    - Add Logging service (e.g. elastic logstash)
    - Test and fix possible bugs
- **For a larger scale implementation**, server side should be splitted into decoupled, independent micro-services, which will communicate through a distributed events bus (e.g. Apache Kafka, RabbitMQ) would allow replication and distribution in several hosts:
    - Collector services
    - Data Transformation services
    - Data Analysis services
    - Data Storage Services
    - etc

Use Case

Using our application we collected tweets on the keywords and usernames lists mentioned in the following paragraphs, and applied basic statistics, listed under the Results paragraph.

Keywords List

- Cultural Heritage
    - tangible
    - intangible

- Digital Cultural Heritage
    - digitization
    - knowledge
    - story
    - memory
    - archiving
    - preservation
    - metadata
    - semantics
    - ontologies
    - Linked data
    - History
    - DigitalHeritage
    - heritagedigitization
    - digitalculturalheritage
    - digitalculture
    - DigitalArchaeology
    - digitalhumanities
    - archaeology
    - archeologia
    - CrowdSourcing
    - EuropeForCulture
    - EuropeanHeritageAlliance
    - CommonCulture
    - CulturalProperty
    - EuropeanHeritageAwards
    - heritage
    - ReligiousHeritage
    - livingheritage

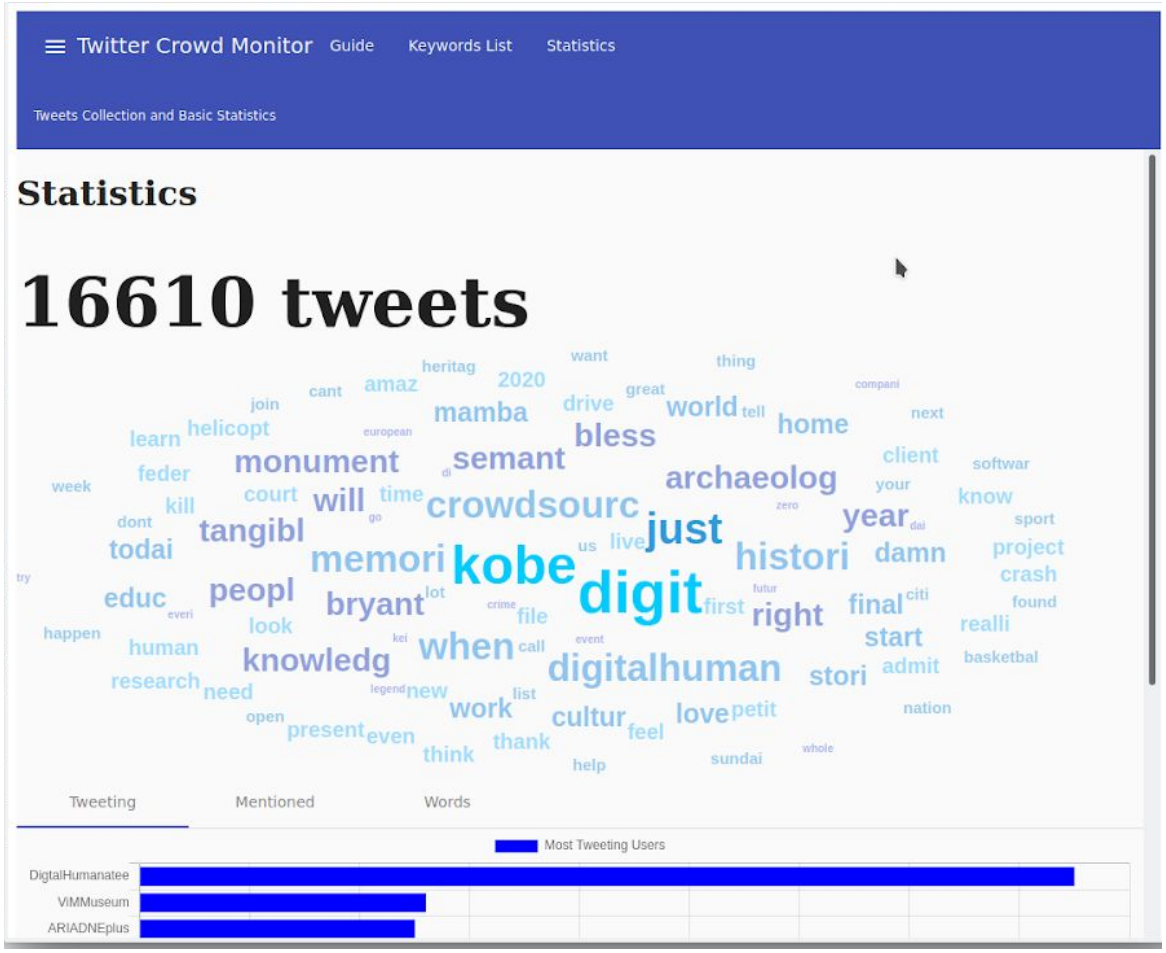
- ViMMuseum
- Unite4Heritage
- Unesco_DCH
- Europeanheritage
- EuropeanCulture
- HeritageProtection
- ProtectHeritage
- intangibleheritage
- intangibleculturalheritage
- tangibleculturalheritage
- tangibleheritage
- heritagepreservation
- CulturalHeritagePreservation
- WorldMonumentsWatch
- EnrichEuropeana
- monument
- digitalarchive
- Europeana Data Model
- CIDOC-CRM
- 3Dheritage
- UNESCO UNITWIN

User Names List

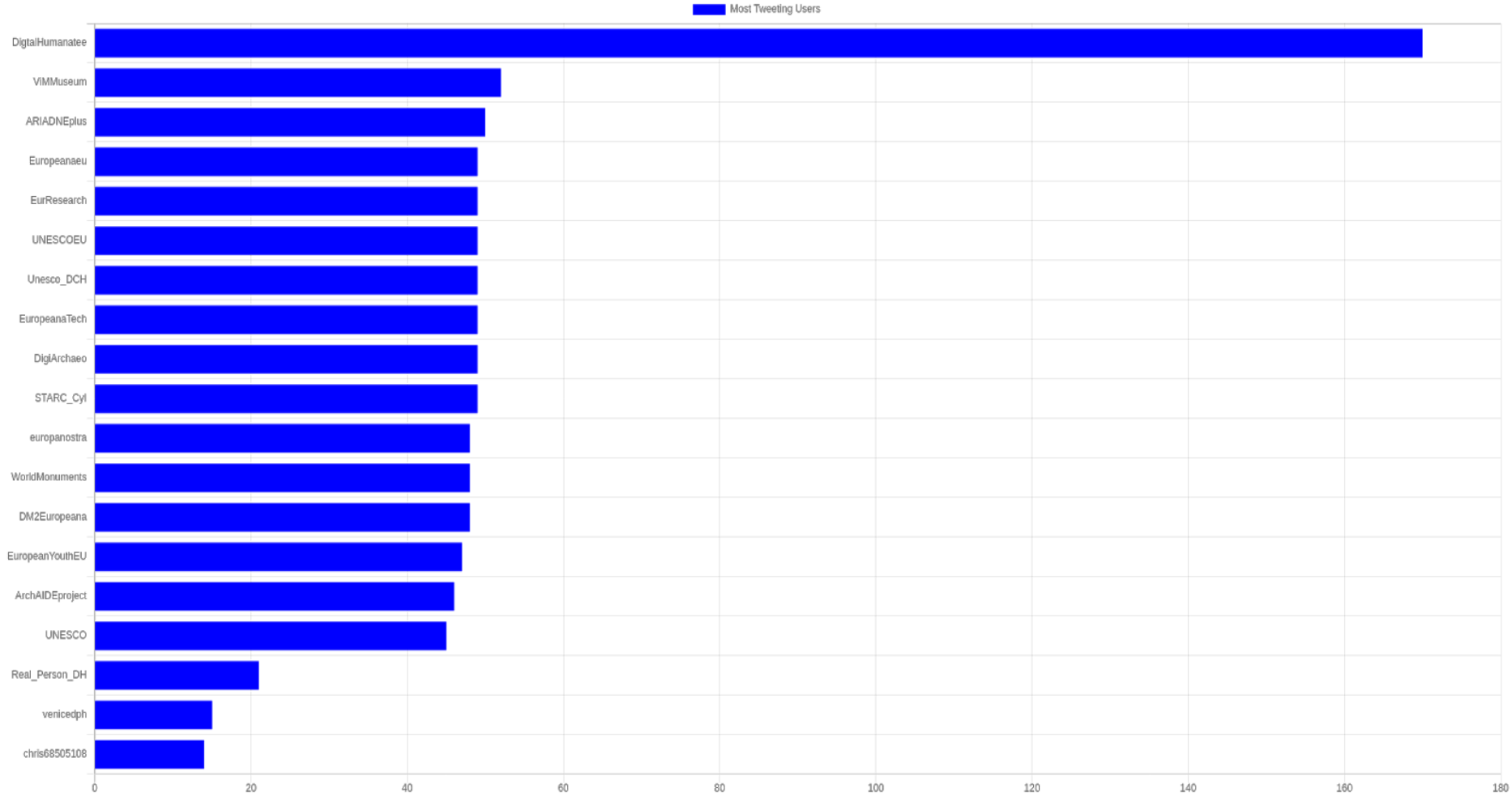
- @UNESCOEU
- @Europeanaeu
- @WorldMonuments
- @Unesco_DCH
- @ViMMuseum
- @europanostra
- @EuropeanYouthEU
- @EuropeanaTech
- @EurResearch
- @ARIADNEplus
- @ArchAIDEproject
- @DigiArchaeo
- @STARC_CyI
- @DM2Europeana

Results

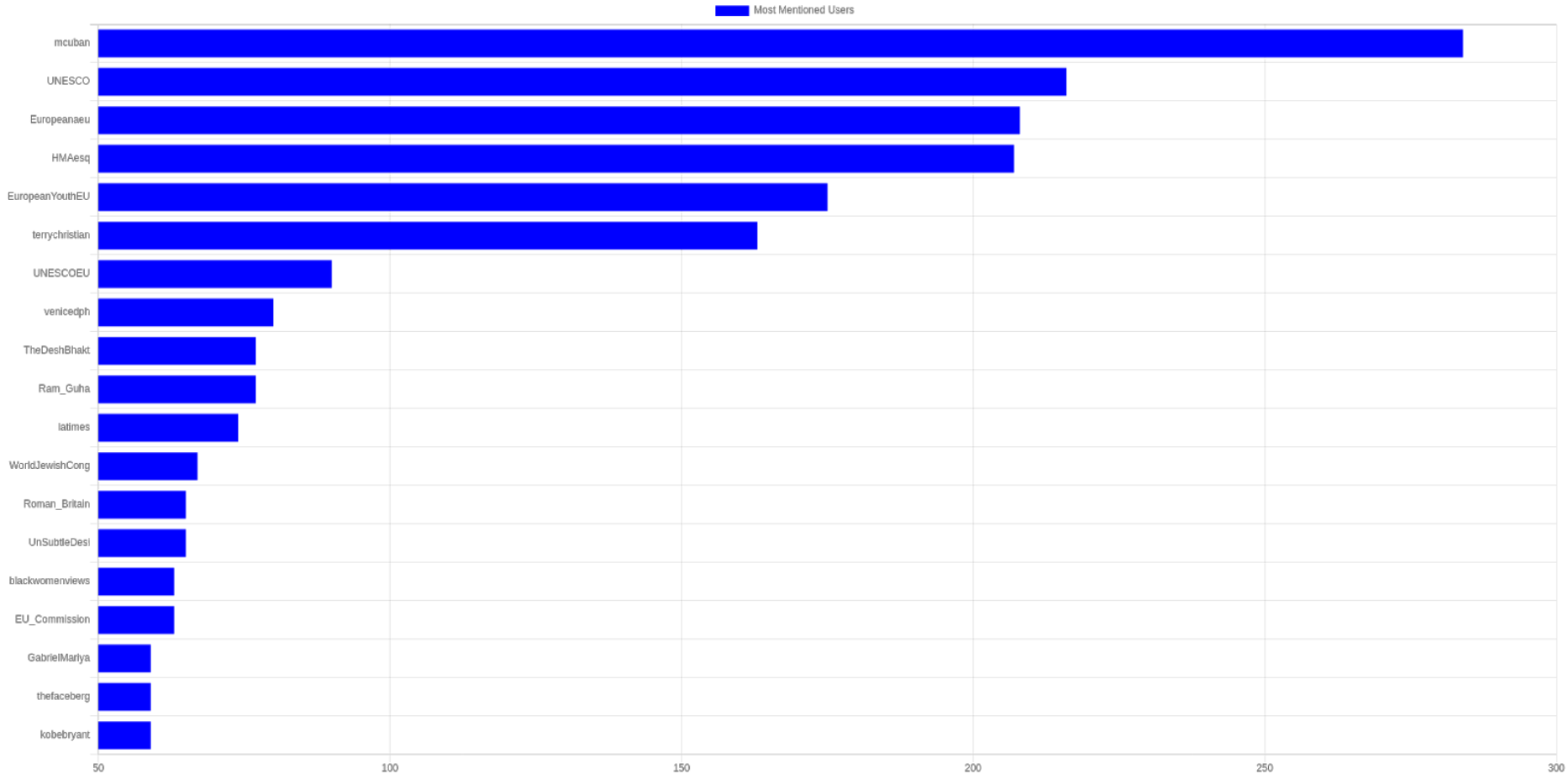
Words Cloud and Total Tweets



Most Tweeting Users



Most Mentioned Users



Most Used Words